

Tools and Methods for the Registration and Fusion of Remotely Sensed Data

Arthur Ardeshir Goshtasby
(Wright State University)

Jacqueline Le Moigne
(NASA Goddard Space Flight Center)

Introduction and background

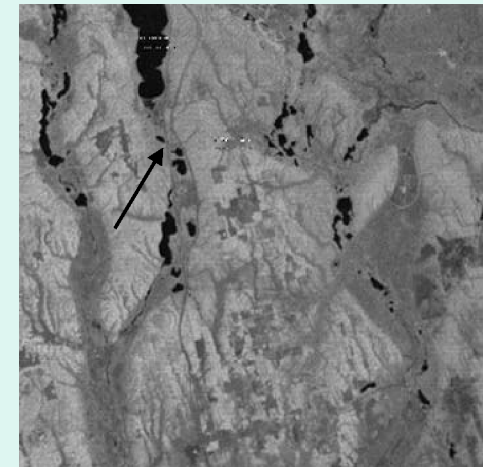
A digital image: An array of scalars or vectors.

Scalar: *Reflectance, temperature, range*

Vector: *RGB, multispectral, hyperspectral*

126	132	124	120	126	124	116	132	126	106	100	104	122	130	120	108
130	136	124	124	124	126	126	132	128	114	104	104	126	136	122	112
130	136	132	126	104	108	122	122	126	120	108	112	128	130	118	112
132	132	128	84	42	40	54	82	112	118	108	118	136	134	114	114
130	132	132	70	4	0	10	32	64	102	116	116	134	130	114	114
128	134	136	102	44	20	16	10	22	78	116	108	124	120	116	114
132	132	136	128	102	60	20	10	22	60	108	108	120	120	112	110
128	126	124	122	124	110	78	48	34	50	100	98	90	118	122	116
122	126	120	114	122	132	128	108	90	86	106	100	84	114	120	114
126	134	130	124	124	124	124	136	140	134	120	110	110	110	106	102
138	138	136	128	124	124	132	130	132	136	124	106	114	114	108	104

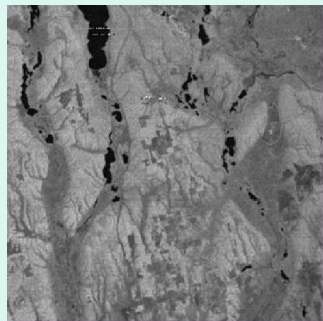
A digital image



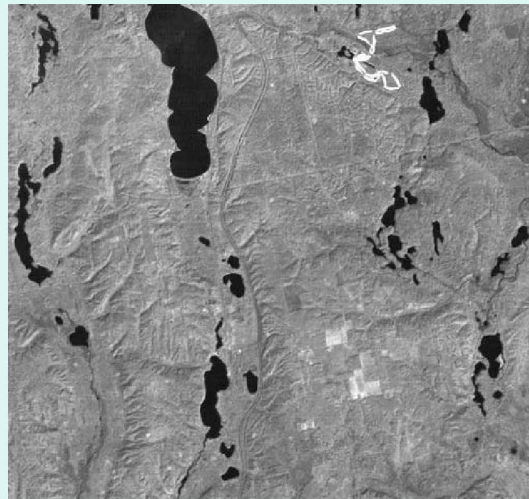
Landsat MSS image,
courtesy of NASA

Image registration and image fusion

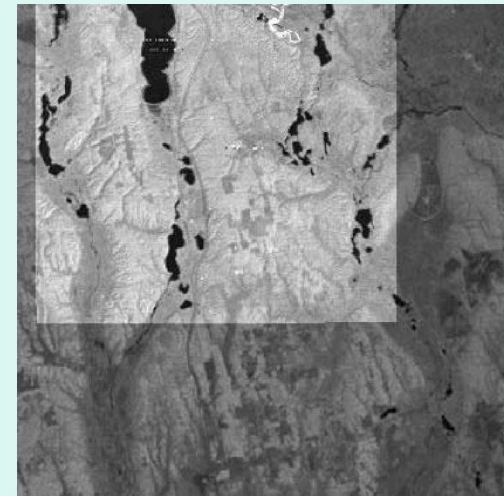
Image registration is the process of spatially aligning two or more images of a scene. This spatial alignment is needed to fuse information in the images.



Landsat MSS



Landsat TM

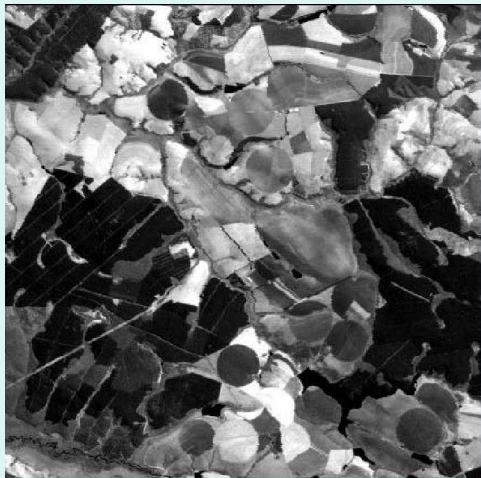


Registered MSS & TM

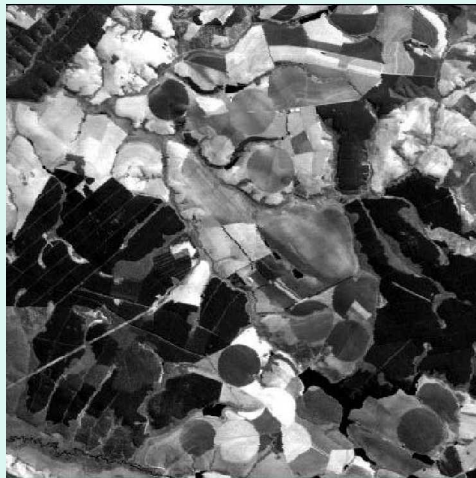
Data courtesy of NASA

Applications of image registration and image fusion

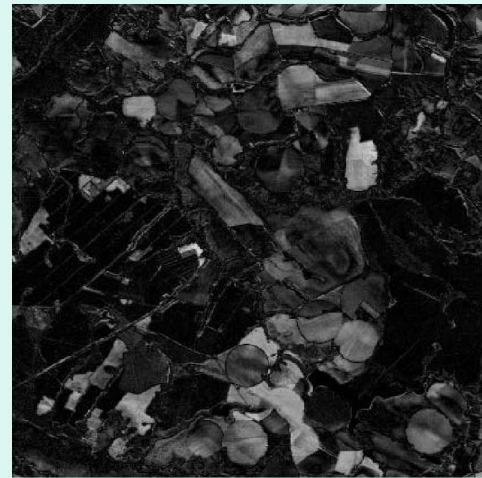
Change detection



Landsat 1



Landsat 2



Change image

Data courtesy of NASA

Fusion of multimodal data



Fused image

Data courtesy of NASA



Landsat TM bands 1 & 7

Image mosaicking



Mosaicked image



Two aerial images of Honolulu, HI.

Outline

1:30–1:40	Introduction and background
1:40–2:00	Landmark detection methods
2:00–2:20	Similarity measures
2:20–2:50	Image features and descriptor
2:50–3:10	The correspondence problem
3:10–3:30	Transformation functions
3:30–3:55	Break

Outline

- 3:55–4:10 Need for fast and accurate methods for registration of remotely sensed data
- 4:10–4:25 Challenges in registration of remotely sensed data
- 4:30–4:45 Image registration at NASA Goddard Space Flight Center
- 4:45–5:00 Fusion of remotely sensed data
- 5:00–5:15 Image fusion via cokriging
- 5:15 – 5:25 Image fusion via image compositing
- 5:25 – 5:30 Concluding remarks

Image Registration: **Preprocessing Operations**

Arthur Goshtasby

Wright State University

Image Registration and Fusion Systems

Preprocessing operations

All operations performed on images that improve the registration performance. These include:

- Noise filtering
- Deblurring
- Region extraction
- Edge detection

Noise smoothing

Given image $f(x,y)$ and smoothing filter $h(i,j)$, noise smoothing is defined by:

$$\bar{f}(x, y) = \sum_{i=-k}^k \sum_{j=-l}^l f(x + i, y + j) h(i, j)$$

The intensity of pixel (x,y) in the output is obtained from a weighted sum of intensities of pixels at and around (x,y) in the input. $h(i,j)$ is the weight of pixel $(x+i,y+j)$ in the neighborhood of (x,y) , and the sum of the weights over all i and j is 1.

Mean filtering

- When the weights defined by the filter are all the same, the operation is known as mean filtering.
- Intensities of all pixels at and around a point in input have the same effect on the intensity at the same point in output.
- This operation is not rotationally invariant if the filter kernel is not circular.

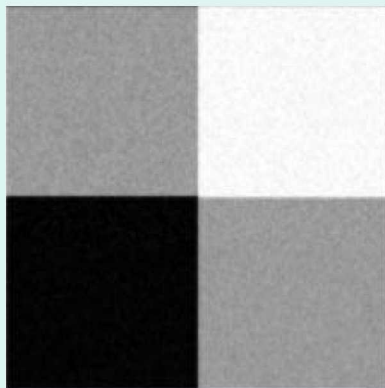
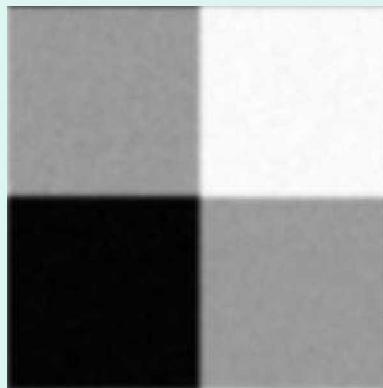
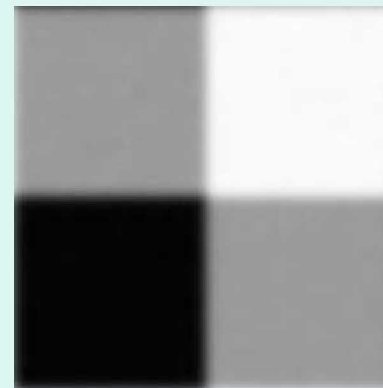


Image containing
Zero-mean noise



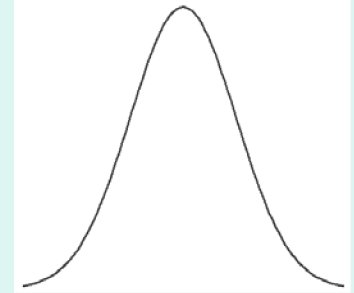
Filter-radius 2 pixels
Computed using FFT algorithm



Filter-radius 4 pixels

Gaussian filtering

- If the weights in filter kernel represent Gaussian coefficients, a Gaussian filter is obtained.
- Gaussian filtering is effective when image noise is zero-mean.
- Gaussian filtering is rotationally invariant.
- A 2-D Gaussian can be decomposed into 2 1-D Gaussians: $G(x,y) = G(x) * G(y)$; therefore, filtering can be carried in 1-D rather than in 2-D



Computation of mean and Gaussian filtering

Although filtering is a convolution operation and can be computed using the FFT algorithm, since FFT considers an image is a periodic signal, if left and right image borders, or top and bottom image borders are not the same, artifacts will appear near the image borders. To avoid this, carry out the computations directly.

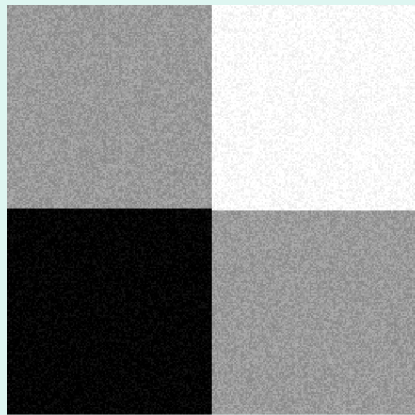
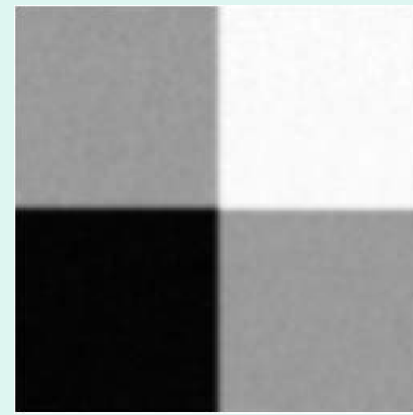


Image containing
Zero-mean noise



Computed with FFT



Computed directly

Gaussian filter of $\sigma = 2$ pixels

Image segmentation

This is the process of partitioning an image into meaningful parts. There are two main approaches to image segmentation.

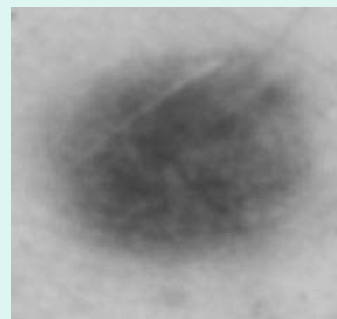
- Methods that use information within regions
- Methods that use information on the boundary between regions

Intensity thresholding

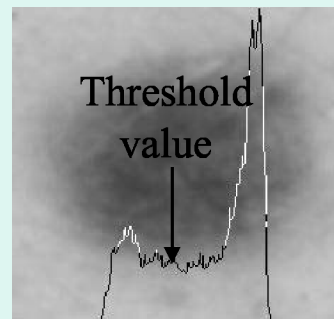
- Assuming an image that contains objects with intensities that represent a Gaussian distribution and a background with intensities that represent a different Gaussian distribution, the objects can be separated from the background using the intensity at the valley between the two histogram modes.
- This method works well when an image contains homogeneous objects and a homogeneous background and the properties of the objects and the background are different.



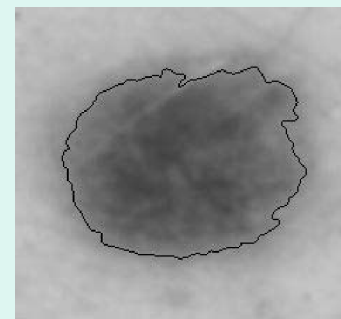
Original



Smoothed

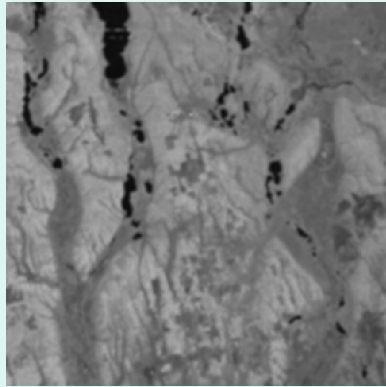


Histogram

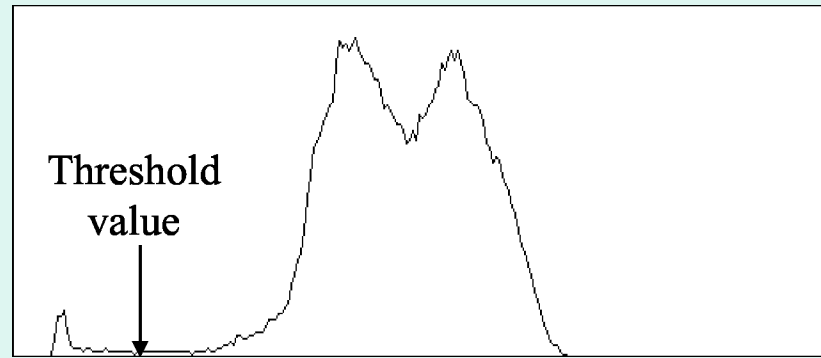


Thresholded image

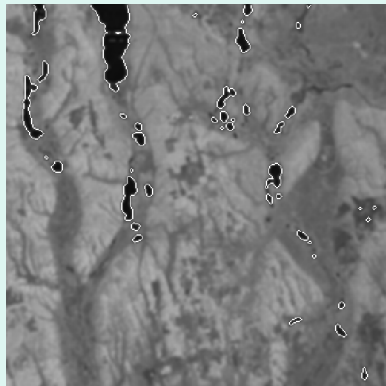
Intensity thresholding



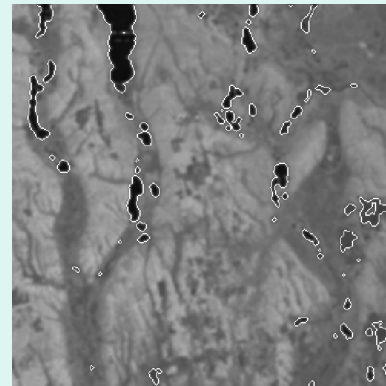
A Landsat image



Histogram of the image



Thresholding at the valley between the first two peaks.



Thresholding at the average intensity of highest-gradient pixels.

Threshold selection

- The threshold value is computed by:
 - Finding the valley between the modes of the histogram of the image.
 - Finding the intensity that represents the average of intensities of high-gradient pixels.
 - Finding the intensity at which a change in the intensity will minimally change the segmentation result.

Edge detection

- Edge detection methods can be categorized into those that search for locally maximum image gradient magnitudes and those that search for zero-crossings of the Laplacian of an image.
- Methods that search for gradient peaks do not pick false edges but the ones picked could be disconnected.
- Methods that search of the zero-crossings of the Laplacian image find closed boundaries, but parts of the boundaries could be false.

LoG edge detector

Determination of the LoG of an image involves computation of:

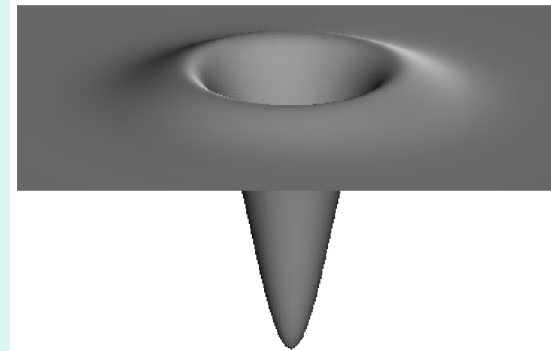
$$\begin{aligned} LoG[f(x, y)] &= \frac{\partial^2 [f(x, y) \star G(x, y)]}{\partial x^2} + \frac{\partial^2 [f(x, y) \star G(x, y)]}{\partial y^2} \\ &= f(x, y) \star \frac{\partial^2 G(x, y)}{\partial x^2} + f(x, y) \star \frac{\partial^2 G(x, y)}{\partial y^2} \end{aligned}$$

or,

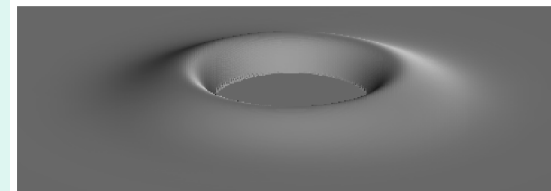
$$LoG[f(x, y)] = \frac{\partial^2 G(x)}{\partial x^2} \star G(y) \star f(x, y) + G(x) \star \frac{\partial^2 G(y)}{\partial y^2} \star f(x, y)$$

Edge detection by intensity ratio

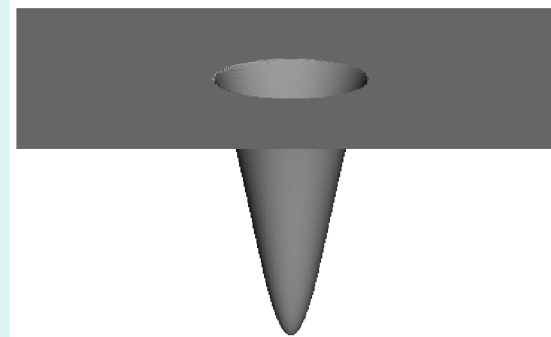
- Zero-crossing edges can be determined by convolving the negative and positive parts of the LoG with an image separately and subtracting the convolved images and locating the zero-crossings.
- If instead of subtracting corresponding values in the convolved images, we divide the values and locate the one-crossings, intensity-ratio edges will be obtained.



LoG
operator in
2-D



Positive
part



Negative
part

Edges in color images

- Edge detection in a color image can be considered edge detection in a 2-D vector field.
- If $\mathbf{u}(x,y)$ and $\mathbf{v}(x,y)$ represent color gradients in x and y directions, edges can be considered points where color gradients are locally maximum in the gradient direction.
- If $R(x,y)$, $G(x,y)$, and $B(x,y)$ represent red, green, and blue color components at (x,y) , respectively, color gradients are:

$$\begin{aligned}\mathbf{u}(x,y) &= \frac{\partial R(x,y)}{\partial x} \mathbf{r} + \frac{\partial G(x,y)}{\partial x} \mathbf{g} + \frac{\partial B(x,y)}{\partial x} \mathbf{b} \\ \mathbf{v}(x,y) &= \frac{\partial R(x,y)}{\partial y} \mathbf{r} + \frac{\partial G(x,y)}{\partial y} \mathbf{g} + \frac{\partial B(x,y)}{\partial y} \mathbf{b}\end{aligned}$$

\mathbf{r} , \mathbf{g} , and \mathbf{b} are unit vectors along red, green, and blue axes, respectively, in the color space.

Color edges

- Gradient direction at (x,y) is the direction maximizing

$$F(x, y) = [\mathbf{u}(x, y) \cos \theta(x, y) + \mathbf{v}(x, y) \sin \theta(x, y)]^2$$

and is obtained from

$$\theta(x, y) = 0.5 \tan^{-1} \left(\frac{2\mathbf{u}(x, y) \cdot \mathbf{v}(x, y)}{\mathbf{u}(x, y) \cdot \mathbf{u}(x, y) - \mathbf{v}(x, y) \cdot \mathbf{v}(x, y)} \right)$$

Edge detection in color images



A color image



Edges of the color image

Related references

1. D. Marr and E. Hildreth, Theory of edge detection, *Proc. R. Soc. Lond.*, **207**:187–217 (1980).
2. J. J. Clark, Authenticating edges produced by zero-crossing algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(1):43–57 (1989).
3. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**:679–714 (1986).
4. L. Ding and A. Goshtasby, On the Canny edge detector, *Pattern Recognition*, **34**:721–725 (2001).
5. A. Goshtasby and H-L Shu, Edge detection by curve fitting, *Image and Vision Computing*, **13**(3): 169–177, 1995.
6. A. Goshtasby, On edge focusing, *Image and Vision Computing*, **12**(4): 247–256, 1994.
7. L. Zagorchev, A. Goshtasby, and M. Satter, R-snakes, *Image and Vision Computing*, **25**: 945–959, 2007.

Landmark Detection

Arthur Goshtasby

Wright State University and
Image Registration and Fusion Systems

Image features

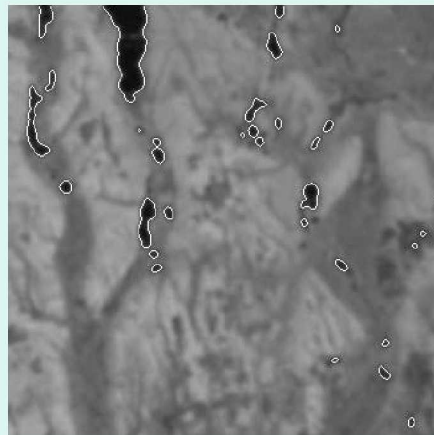
Points



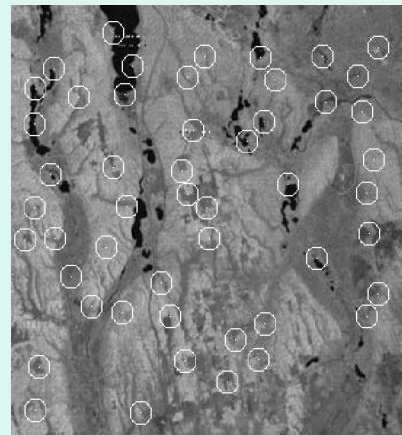
Lines



Regions



Templates



Landmark detection

- They are 1) locally unique and 2) rotationally invariant.



Corners detected at different resolutions.

Corners as landmarks

Given image I :

1. Compute image gradients in x and y directions: I_x , I_y
2. Compute square gradient matrix at each pixel (x,y) :

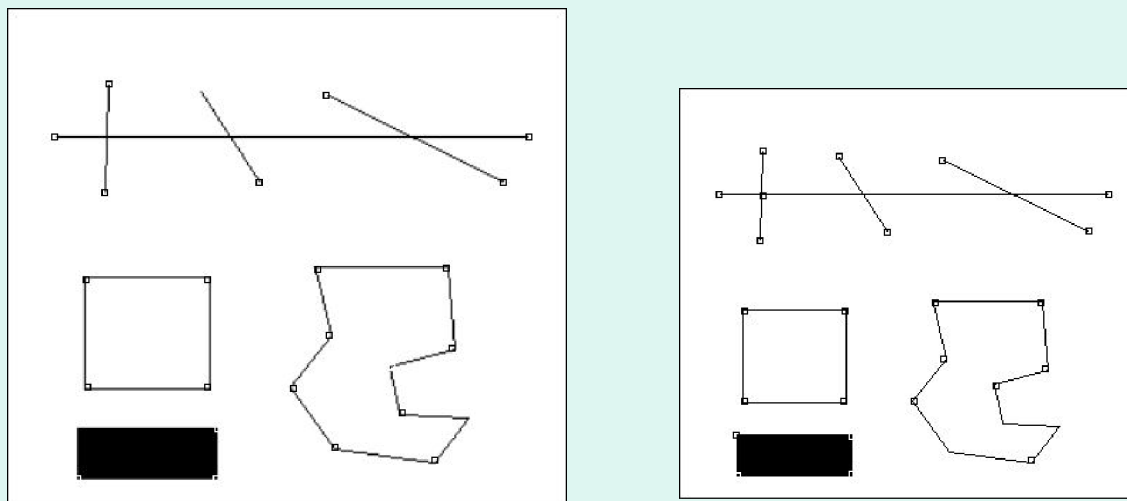
$$\mathbf{C}(x, y) = \begin{bmatrix} \overline{I_x I_x} & \overline{I_x I_y} \\ \overline{I_y I_x} & \overline{I_y I_y} \end{bmatrix}$$

where overbar implies average in a small neighborhood of (x,y) .

3. Compute eigenvalues of $\mathbf{C}(x,y)$ and if the smaller eigenvalue is locally maximum, take (x,y) as a corner.

Examples

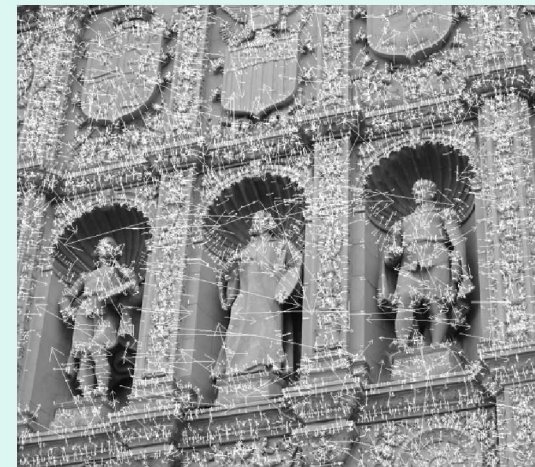
Stable corners: Corners that persist over a wide range of scales/resolutions.



Corners detected at two different scales.

Spot centers as landmarks

- Find points where Laplacian of Gaussian (LoG) of image is locally extremum.
- Scale-invariant feature transform (SIFT): LoG points that are locally extremum in scale and space.



Locally unique and highly informative points as landmarks

Given an image and a threshold distance d :

1. Find edges in the image.
2. At each edge point compute entropy within a circular window centered at it.
3. Remove edge points that have produced the lowest $p\%$ entropies.
4. Compute the auto-correlation at the remaining edge points and save an edge point in list L , which is initially empty, if the largest auto-correlations there is locally minimum.
5. Sort list L from the smallest to the largest auto-correlation.
6. Consider the edge point on top of the list the next corner.
7. Remove all edge points that are within distance d of the selected edge point from L .
8. If there are still edge points in L , go to Step 6. Otherwise stop.

An Example



Line intersections as landmarks

Line detection

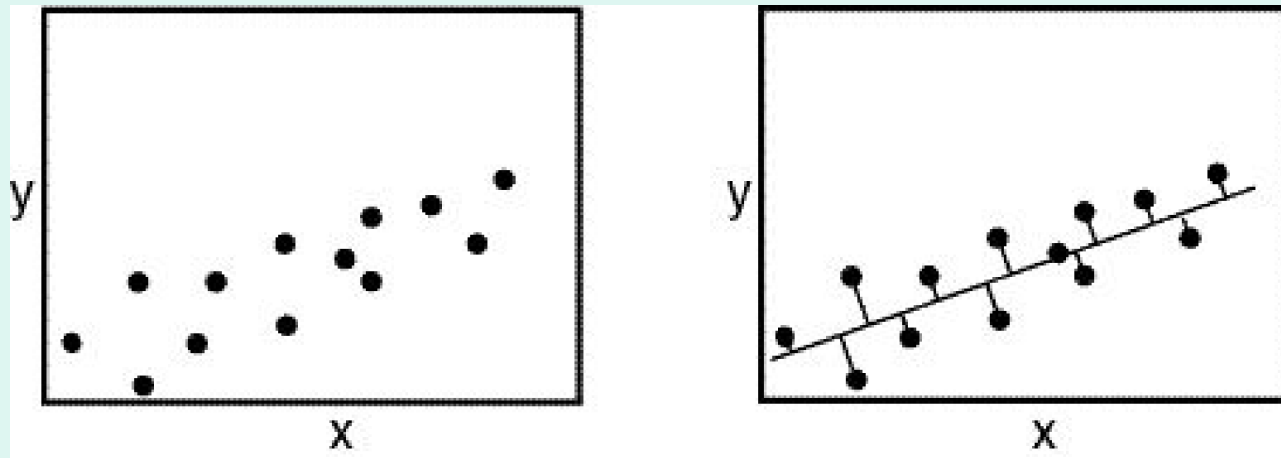
- Least-squares
- Hough transform
- Image gradients

Least-squares line fitting

Given a set of points $\{(x_i, y_i): i=1, \dots, n\}$ find ρ and θ of line $\rho = x \cos \theta + y \sin \theta$ such that

$$E = \sum_{i=1}^n (\rho - x_i \cos \theta - y_i \sin \theta)^2$$

is minimum.



An example



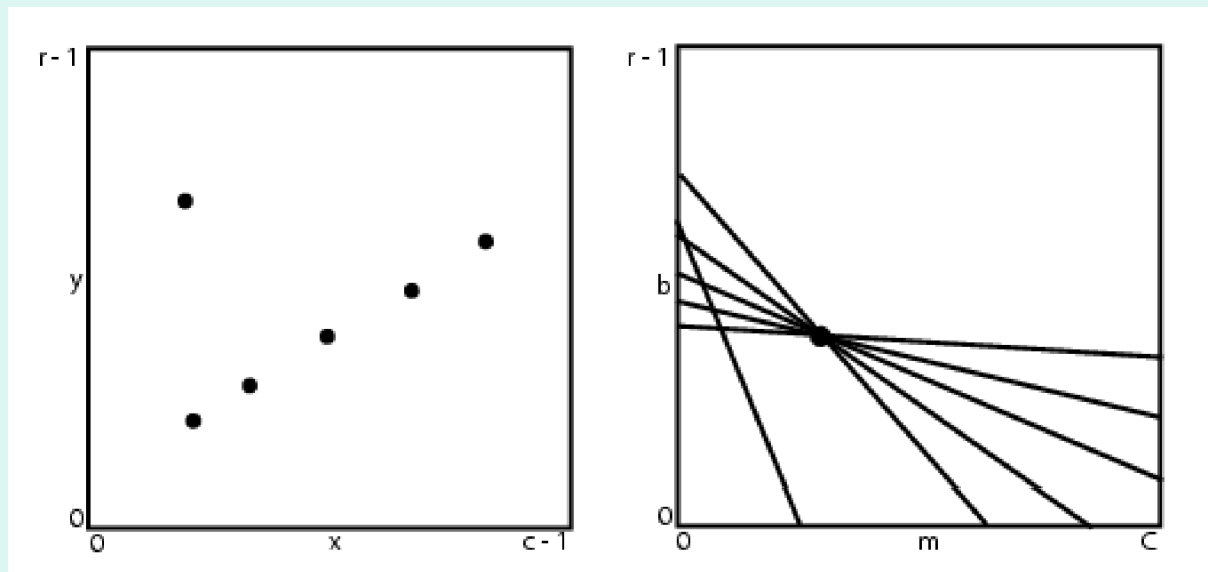
Canny edges



Detected lines

Hough transform

A point (x,y) in the xy -space represents a unique line $y = mx + b$ in the mb -space, and a point (m,b) in the mb -space represents to a unique line in the xy -space.



Hough transform

Parameter m cannot be accurately determined for nearly vertical lines. So, instead of $y=mx+b$, the polar form of line $\rho = x\cos\theta + y\sin\theta$ is used.

Algorithm: Given an edge image:

1. For each edge point (x,y) , draw $\rho = x\cos\theta + y\sin\theta$ in a $\rho\theta$ array. As each sinusoidal is drawn, increment $\rho\theta$ entries in the array that fall on the sinusoid by 1.
2. Locate locally peak entries in the array. Each such entry in the array $\rho\theta$ detects a line in the image.

Line detection using image gradients

1. Determine the gradient magnitude and gradient direction of each pixel in the image.
2. Group the pixels into regions with gradient directions in $[-\alpha, \alpha]$ $[\alpha, 3\alpha]$. . . , where α is a small angle, such as 5 degrees.
3. Remove regions that contain fewer than a required number of pixels.
4. Fit a line to each remaining region by the weighted least-squares method, with the weights being the gradient magnitudes at the pixels.

An example

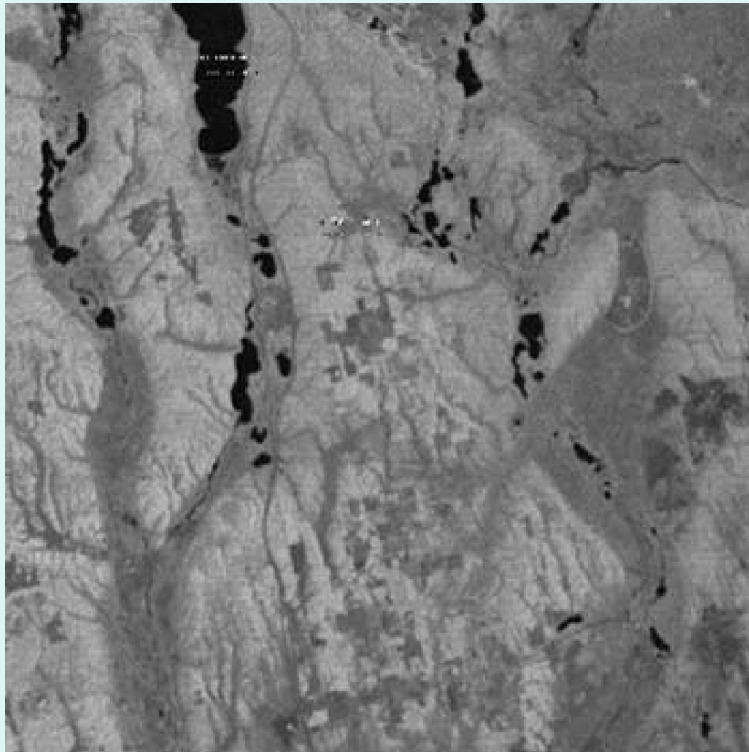


Lines detected at two different values of α .

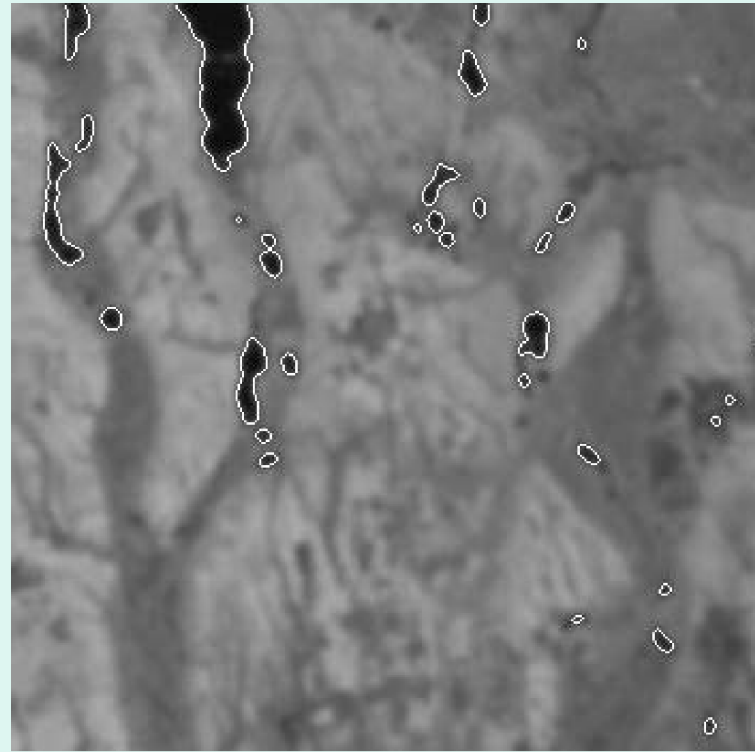
Region centers as landmarks

- For some image types, regions are the best features to use in image registration.
- Centers of gravity of regions are not sensitive to zero-mean noise.
- Centers of gravity of regions can be determined with subpixel accuracy.
- Centers of gravity of regions are affine invariant.
- Regions can be obtained by various image segmentation methods.

An example



Landsat MSS image

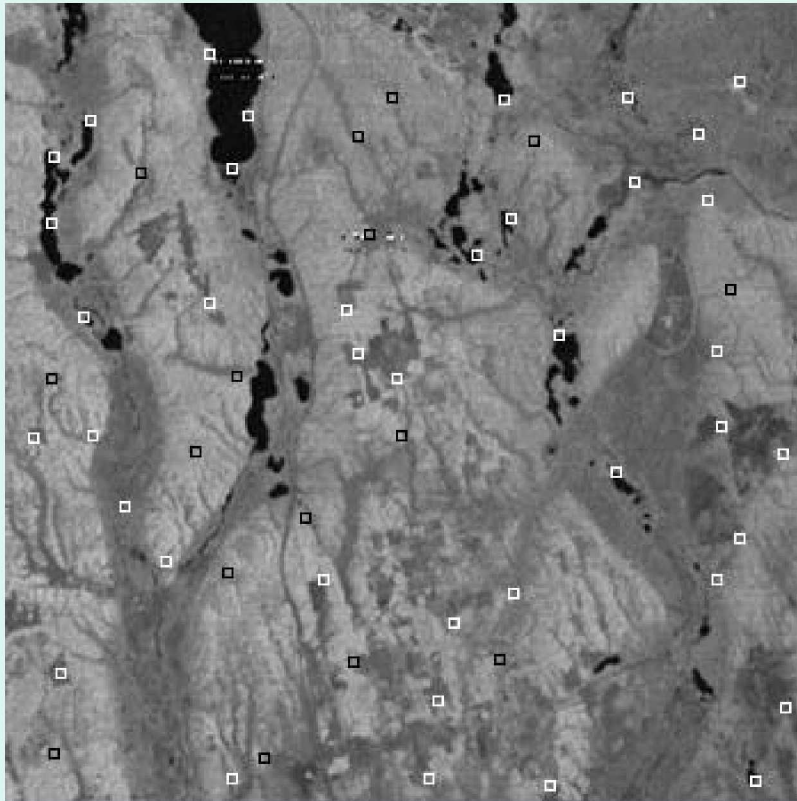


Smoothed and thresholded image

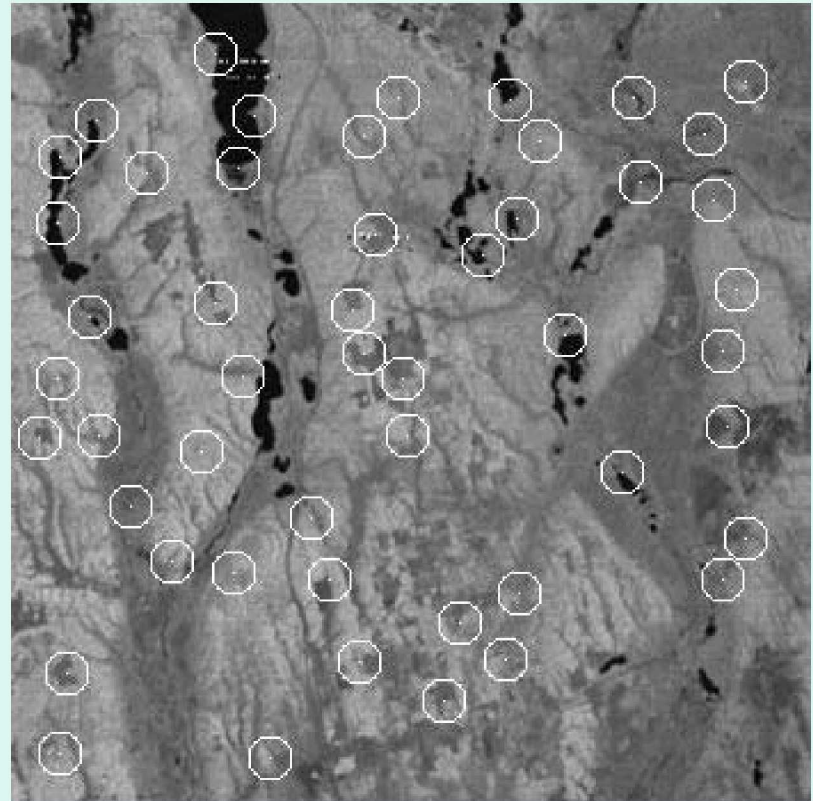
Template centers as landmarks

- Templates are image regions that are locally unique and informative.
- They can be considered circular windows that are centered at detected landmarks.
- Circular windows make the selected features independent of an image's orientation.

An example



Corners



Templates

Related references

1. D. Lowe, Distinctive image features from scale-invariant keypoints, *Int'l J. Computer Vision*, 60(2):91–110, 2004.
2. C. Harris and M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conf. (AVC88)*, 147–151, 1988.
3. J. Illingworth and J. Kittler, A survey of the Hough transform, *Computer Vision, Graphics, and Image Processing*, **44**:87–116 (1988).
4. J. B. Burns, A. R. Hanson, and E. M. Riseman, Extracting straight lines, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**(4):425–455 (1986).
5. A. Goshtasby, G. Stockman, and C. Page, A region-based approach to digital image registration with subpixel accuracy, *IEEE Trans. Geoscience and Remote Sensing*, **24**(3):390–399 (1986).
6. A. Goshtasby, Template matching in rotated images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **7**(3):338–344 (1985).

Similarity Measures

Arthur Goshtasby

Wright State University

Image Fusion Systems Research

Definitions

- A measure to quantify how well two patterns or two image windows match.
- Dissimilarity measure is also known as distance measure.
- A similarity/dissimilarity measure may or may not represent a metric.
- If SD is a metric similarity/dissimilarity, then $SD(X,Y) = SD(Y,X)$, for windows X and Y.

Pearson correlation coefficient

Given intensities $X = \{x_i : i = 1, \dots, n\}$ and $Y = \{y_i : i = 1, \dots, n\}$,

$$r = \frac{1}{n} \sum_{i=1}^n \left(\frac{(x_i - \bar{x})}{\sigma_x} \right) \left(\frac{(y_i - \bar{y})}{\sigma_y} \right)$$

r varies between +1 and -1. +1 shows perfect positive correlation and -1 shows perfect negative correlation.

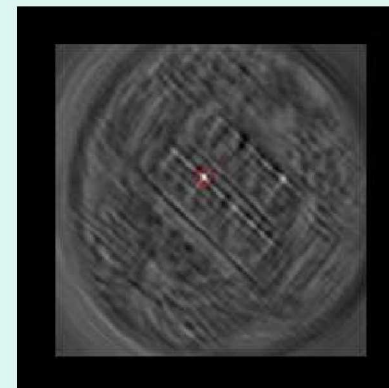
Computation of correlation

- Using Fourier transform

$$C = \mathcal{F}^{-1} [\mathcal{F}(U) \cdot \mathcal{F}^*(V)]$$

- Phase component of cross-power spectrum

$$C_p = \mathcal{F}^{-1} \left[\frac{\mathcal{F}(U) \cdot \mathcal{F}^*(V)}{|\mathcal{F}(U) \cdot \mathcal{F}^*(V)|} \right]$$



r



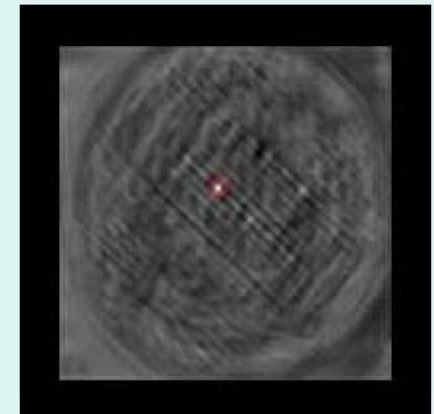
C_p

Spearman's rho

- Suppose replacing intensities x_i and y_i with their ranks $R(x_i)$ and $R(y_i)$ and then calculating the Pearson correlation coefficient between the ranks. This is equivalent to finding:

$$\rho = 1 - \frac{6 \sum_{i=1}^n [R(x_i) - R(y_i)]^2}{n(n^2 - 1)}$$

- This measure is invariant to monotone intensity differences between images.

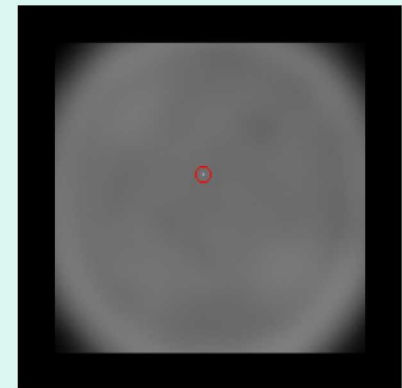


ρ

Shannon's mutual information

- Assuming p_{ij} is the probability that corresponding pixels in X and Y have intensities i and j , mutual information is defined by:

$$S_{MI} = \sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij} \log_2 \frac{p_{ij}}{p_i p_j}$$



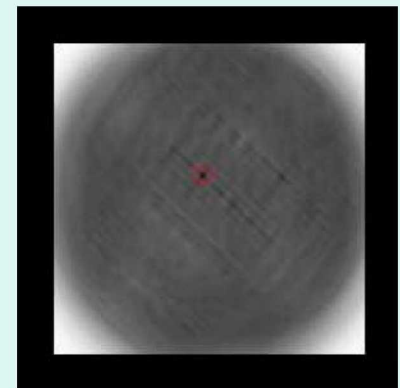
L₁ and L₂ norms

- L₁ norm is also known as sum of absolute intensity differences.

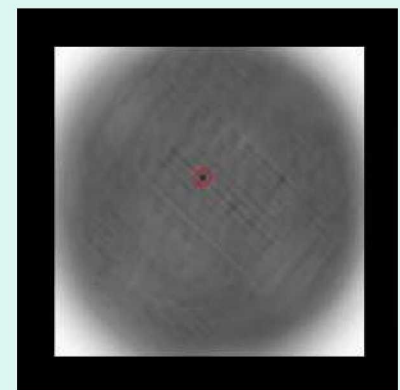
$$L_1 = \sum_{i=1}^n |x_i - y_i|$$

- L₂ norm is also known as the sum of squared intensity differences.

$$L_2^2 = \sum_{i=1}^n (x_i - y_i)^2$$



L₁

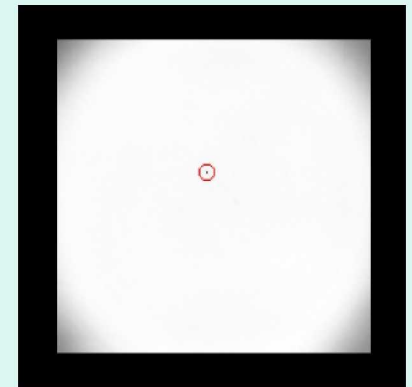


L₂

Joint entropy

- Having the joint probability density p_{ij} of intensities i and j in X and Y , joint entropy is defined by:

$$D_E = - \sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij} \log_2 p_{ij}$$

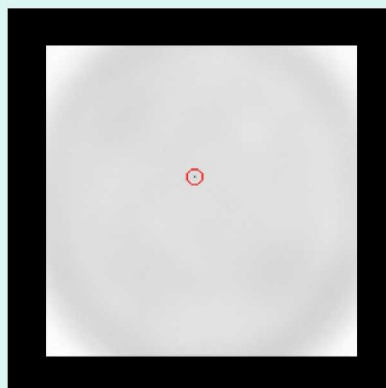


- D_E should be minimized to find correspondence.

Exclusive f -information

- Information exclusively contained in X and Y when observed jointly is known as f -information.
- Exclusive f -information is related to joint entropy D_E and mutual information S_{MI} by:

$$D_f(X, Y) = D_E(X, Y) - S_{MI}(X, Y)$$



Properties

- Sensitivity to noise: Spearman's ρ and L_1 norm.
- Sensitivity to intensity differences: Mutual information, joint entropy, and Spearman's ρ .
- Sensitivity to blurring: Pearson's correlation coefficient, L_1 norm, and L_2 norm.
- Sensitivity to geometric difference: Pearson's correlation coefficient and L_1 norm.
- Speed: L_1 norm, L_2 norm, Pearson's correlation coefficient.

Related references

- D. I. Barnea and H. F. Silverman, A class of algorithms for fast digital image registration, *IEEE Trans. Computers*, vol. 21, no. 2, pp. 179-186, 1972.
- W. Kruskal, Ordinal measures of association, *Journal of American Statistical Association*, vol. 53, pp. 814-861, 1958.
- C. D. Kuglin and D. C. Hines, The phase correlation image alignment method, *Proc. Int'l Conf. Cybernetics and Society*, pp. 163-165, 1975.
- F. Maes, D. Vandermeulen, and P. Suetens, Medical image registration using mutual information, *Proc. IEEE*, vol. 91, no. 10, pp. 1699-1722, 2003.
- K. Pearson, Contributions to the mathematical theory of evolution, III, Regression, heredity, and panmixia, *Philosophical Transactions Royal Society London, Series A*, vol. 187, pp. 253-318, 1896.
- J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, f -information measures in medical image registration, *IEEE Trans. Medical Imaging*, vol. 23, no. 12, pp. 1506-1518, 2004.
- C. Spearman, The proof and measurement of association between two things, *The American Journal of Psychology*, vol. 15, no. 1, pp. 72-101, 1904.

Image Features

Arthur Goshtasby

Wright State University

Image Fusion Systems Research

Types of image features

- Statistical
- Geometric
- Algebraic
- Frequency domain
- Filter response
- Differential
- Fractal dimension
- Information theoretic

Statistical features

Assuming $p(i)$ is the probability that intensity i appears in image:

Mean:

$$\mu = \sum_{i=0}^{255} ip(i)$$

Variance:

$$\sigma^2 = \sum_{i=0}^{255} (i - \mu)^2 p(i)$$

Skewness:

$$\gamma = \frac{1}{\sigma^3} \sum_{i=0}^{255} (i - \mu)^3 p(i)$$

Kurtosis:

$$\kappa = \frac{1}{\sigma^4} \sum_{i=0}^{255} (i - \mu)^4 p(i) - 3$$

Examples



Image

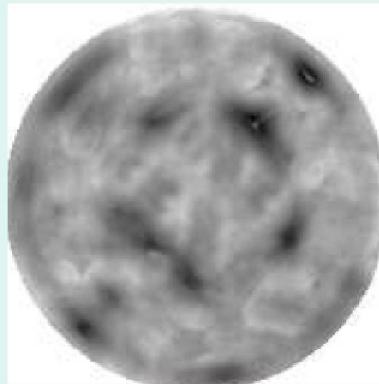


Mean



Variance

Feature obtained when using
windows of radius 8 pixels.



Skewness



Kurtosis

Geometric features

Invariant moments:

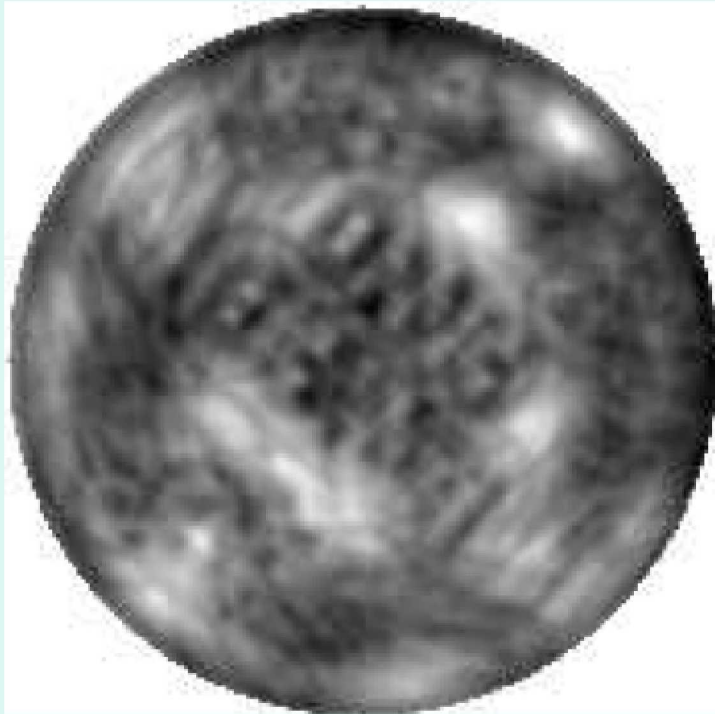
Central moments: Assuming (x_0, y_0) is center of gravity of image $f(x, y)$

$$\mu_{pq} = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (x - x_0)^p (y - y_0)^q f(x, y)$$

Invariant moments: (order 2)

$$I_1 = (\mu_{20} + \mu_{02}),$$
$$I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

Examples



Invariant moment I_1



Invariant moment I_2

Algebraic features

- Given image f , if U is the column eigenvector system of f and V is the row eigenvector system of f , we can write:

$$\mathbf{U}f\mathbf{V} = \begin{pmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, where

$\sigma_i = \lambda_i^{1/2}$ is the i th largest singular value.

Singular values

- Rather than σ_i , use σ_i/σ_1 ; it is invariant to image contrast.

Example:



σ_2/σ_1

Frequency domain features

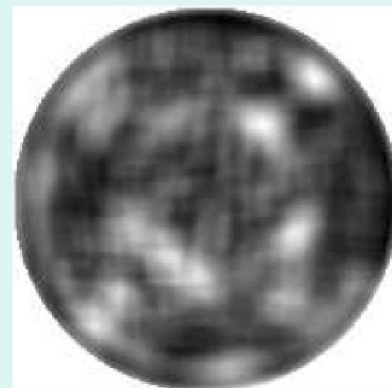
- Power spectrum features: If $F(u, v)$ is the Fourier transform of image $f(x, y)$, then

$$\phi(u, v) = F(u, v)F^*(u, v) = ||F(u, v)||^2$$

are the power spectrum features of the image.

Total power spectrum:

$$\sum_u \sum_v ||F(u, v)||^2$$



Filter responses

- Consider 1-D filters:

$$\begin{array}{lcl} B_0 & = & \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \\ B_1 & = & \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \\ B_2 & = & \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix} \\ B_3 & = & \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix} \\ B_4 & = & \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix} \end{array}$$

and suppose creating 2-D filter $B_{ij}=B_i*B_j$, then response of image to B_{ij} can be used as a feature.

An example

- Response to B_{11} :



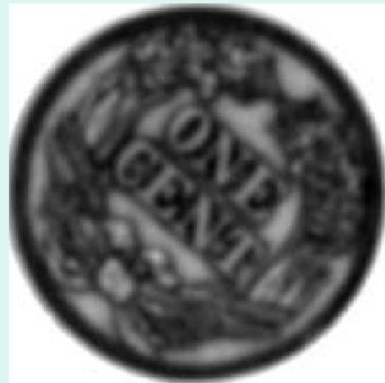
Differential features

- Rotationally invariant differential operators:

Smoothed: $\bar{f}(x, y),$

Gradient Magnitude: $\left\{ \bar{f}_x^2(x, y) + \bar{f}_y^2(x, y) \right\}^{\frac{1}{2}},$

Laplacian: $\bar{f}_{xx}(x, y) + \bar{f}_{yy}(x, y),$



Smoothed



Laplacian

Spatial domain features

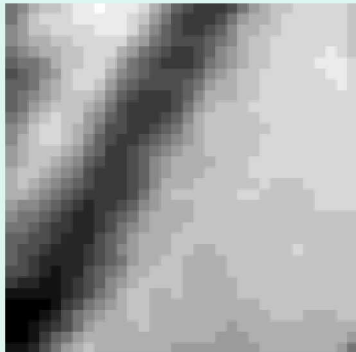
- Absolute center contrast:

$$\frac{1}{MN - 1} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x, y) - f_c|$$

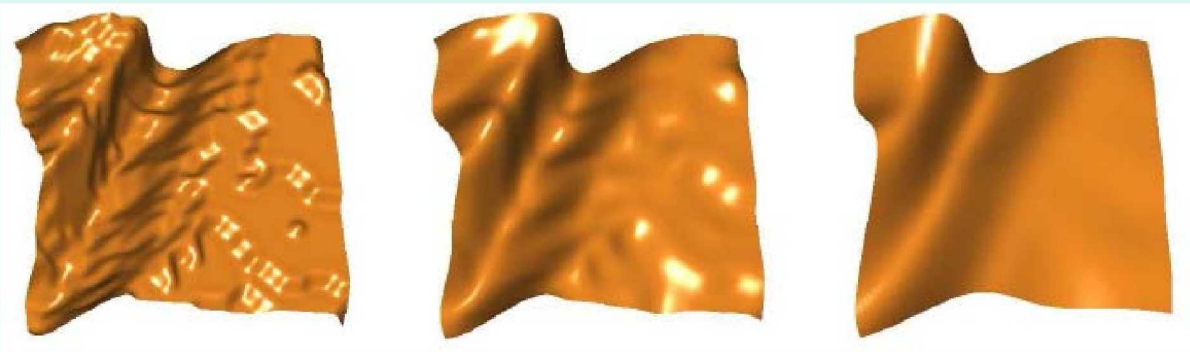


Fractal dimension

- It is a measure of roughness when considering intensities as height values. Change in surface area as a function of change in resolution is used as fractal dimension.

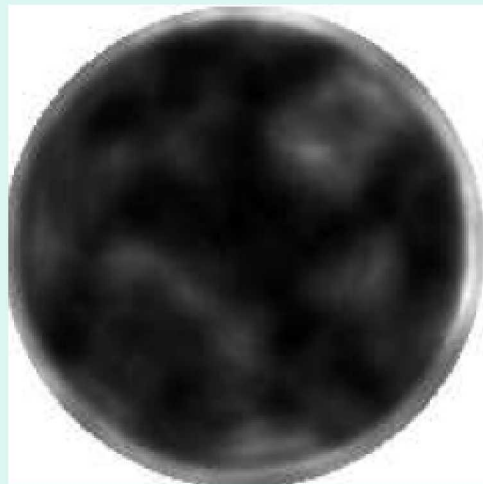


Intensities



Surfaces at different resolutions

An example

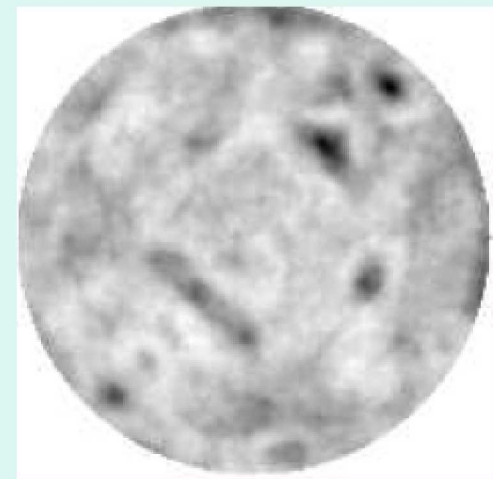


Fractal dimensions

Information theoretic features

- **Entropy:** It measures information content in an image or window.

$$-\sum_{i=0}^{255} P(i) \log_2 P(i)$$



Properties of features

- Invariance under blurring: Invariant moments, smoothed intensity
- Invariance under noise: Smoothed intensity, invariant moments, fractal dimension
- Invariance to change in contrast: Gradient filter response
- Invariance to rotation and scale: Smoothed intensity, invariant moments, Laplacian

Related references

- H. C. Andrews and C. L. Patterson, Singular value decomposition and digital image processing, *IEEE Trans. Acoustics, Speech, and Signal Processing*, pp. 26-53, 1976
- M. K. Hu, Visual pattern recognition by moment invariants, *IEEE Trans. Information Theory*, vol. 8, pp. 179-187, 1962.
- J. J. Koenderink and A. J. van Doorn, Representation of local geometry in the visual system, *Biological Cybernetics*, vol. 55, pp. 367-375, 1987.
- K. I. Laws, Rapid texture identification, in *Image Processing for Missile Guidance*, *Proc. SPIE*, vol. 238, pp. 376-380, 1980.
- A. Pentland, Fractal-based description of natural scenes, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 661-674, 1984.
- C. E. Shannon, The mathematical theory of Communication, in book with the same title by C. E. Shannon and W. Weaver, University of Illinois Press, Urbana, 1949, reprint 1998.

Image Descriptors

Arthur Goshtasby

Wright State University

Image Fusion Systems Research

What is an image descriptor?

It is a feature vector. Traditionally, all features are taken to be of the same type, but this is a self-imposed restriction.

- Histogram-based
- SIFT
- GLOH
- Shape context
- Spin image
- RIFT

Histogram-based descriptors

- A feature vector with components representing the histogram bin counts.
- Assuming H_1 and H_2 histograms of two windows, and the i th bin count in the histograms are $H_1(i)$ and $H_2(i)$, distance between the two histograms is determined from $D(H_1, H_2) = \sum_i \min_i \{H_1(i), H_2(i)\}$.

Scale-invariant feature transform (SIFT)

- Find SIFT points.
- Estimate scale and orientation of neighborhood of each point.
- Normalize a local neighborhood and subdivide into 4×4 blocks.
- Create a gradient histogram for each block.
- Concatenate the local histograms to create the descriptor.

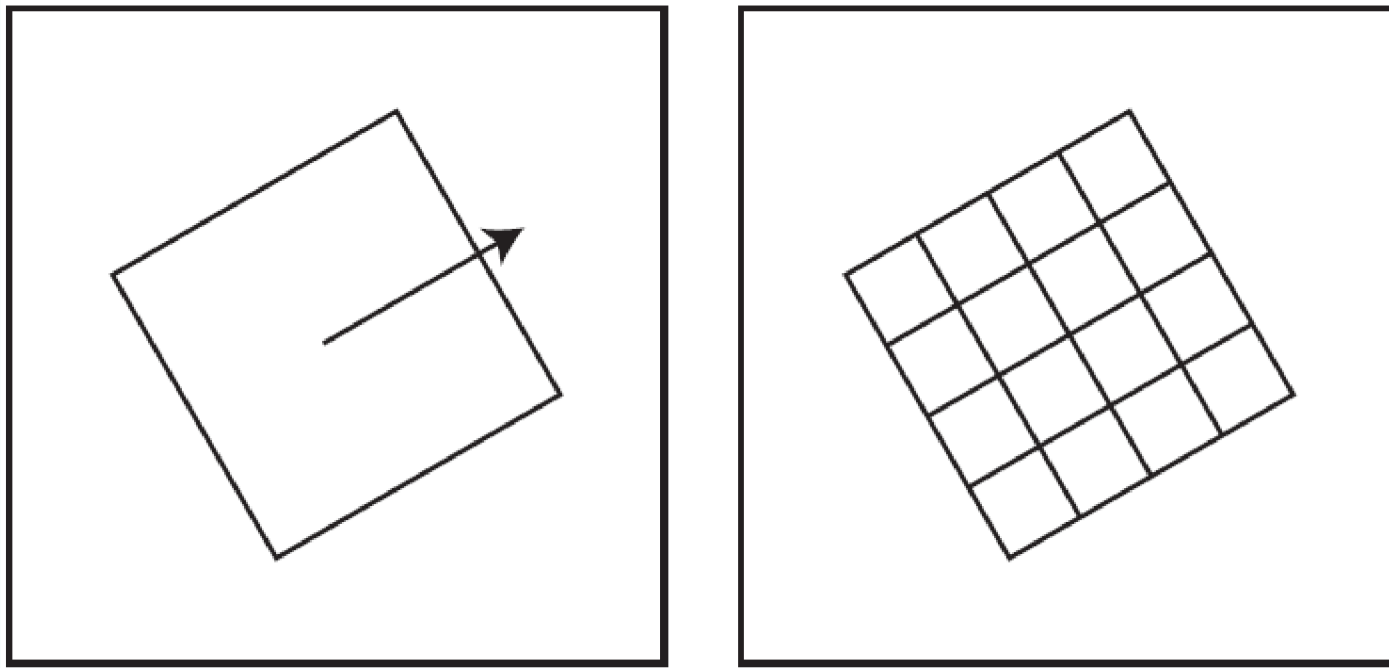
Finding SIFT points

- Convolve image with Gaussians of standard deviations $\sqrt{2}, 2, 2\sqrt{2}, 4 \dots$, creating a stack of Gaussian smoothed images.
- Find difference of Gaussian (DoG) smoothed images in the stack to create another stack, approximating the LoG of the image at different resolutions.
- Consider the stack of the DoG images a volumetric images and find extremum points in the volume.
- Slice number of an extremum determines the scale of neighborhood of the point.

Estimating orientation

- Knowing the slice number, n , of a point in the DoG stack, find the gradient magnitude and gradient direction of slice n in the Gaussian smoothed stack.
- Weigh gradient magnitudes by a Gaussian with standard deviation proportional to n .
- Find histogram of gradient directions within a square neighborhood of width proportional to n , using weighted gradient magnitudes as increments.
- Take the direction representing the histogram peak as the orientation of the point.

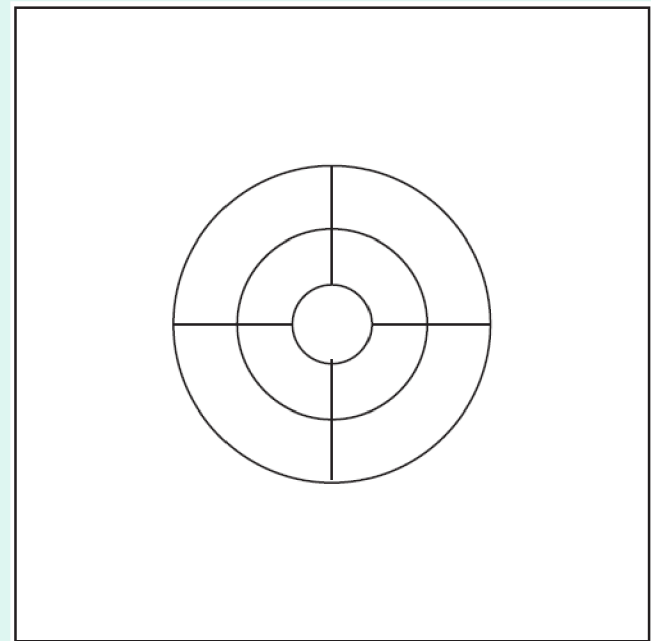
Normalizing and subdividing neighborhood



Gradient directions in each block are normalized with respect to the peak orientation and grouped into a Histogram with 8 bins.

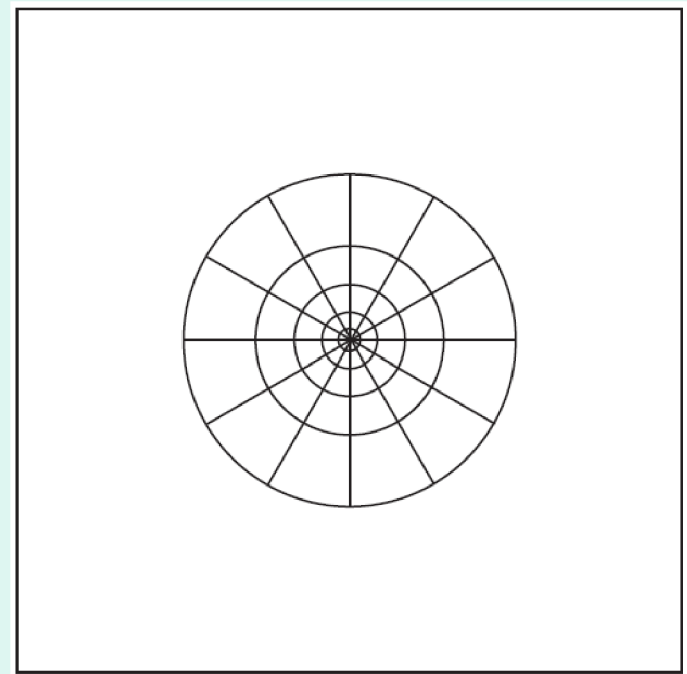
Gradient location and orientation histogram (GLOH)

- Subdivide the neighborhood into a log-polar grid with radii 5, 11, 16, and 90-degree angular spacing.
- Group gradient directions within each block into a histogram with 16 bins.
- Overall, producing 272 numbers, but reducing to 128 by PCA.



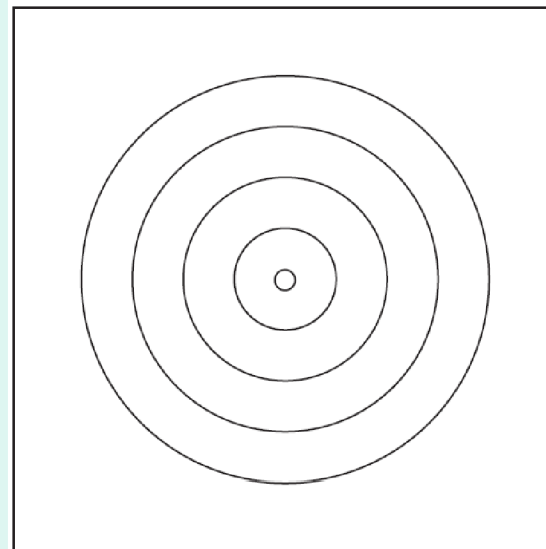
Shape context

- Same as GLOH, but spacing the rings uniformly spaced in log space. This emphasizes the center part of a window.
- Five circular blocks are used and pixels between consecutive rings are grouped into 12 directions, producing overall 60 numbers.

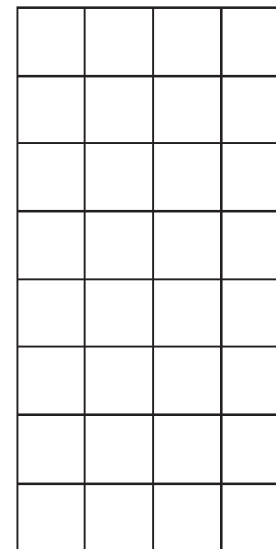


Spin image

- Consider grouping intensities within each circular block into a histogram with 8 bins, and mapping the bin counts to column entries in a matrix. This will produce a rotation-invariant matrix known as spin image.



Image



Spin image

Rotation-invariant feature transform (RIFT)

- If instead of image intensities, gradient directions that are measured radially are used, the obtained spin image has been named RIFT.
- Note that gradient direction measured radially is independent of the orientation of the images, thus creating a rotation invariant descriptor.

How reliable are SIFT rotation and scale parameters?



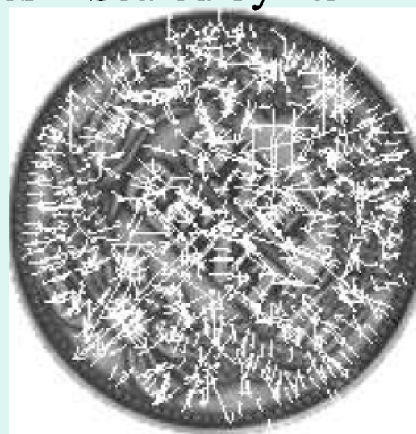
Original



Rotated by 30 degrees



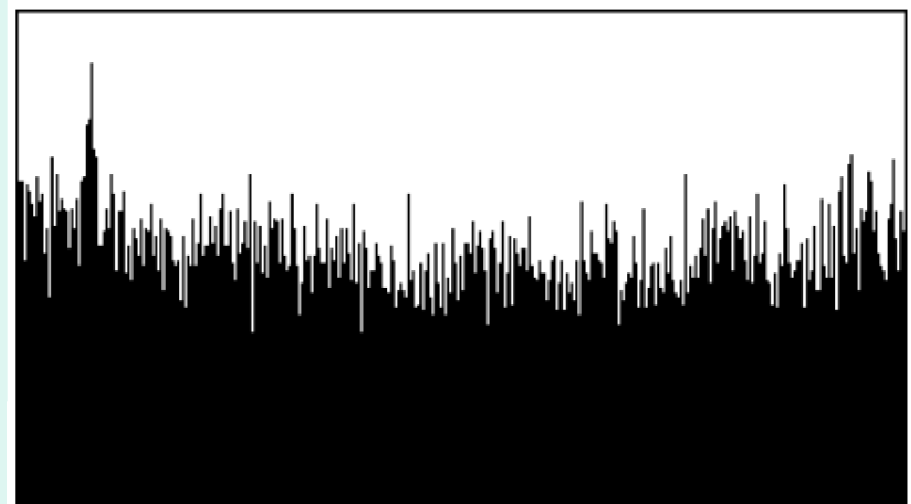
Scaled by 1.5



SIFT points with associating orietations and scales

SIFT rotation parameters

- Find MST of SIFT points before and after rotation.
- Consider points with the same degree in the two MSTs.
- Find difference in SIFT rotations at such points.
- Find histogram of the rotational differences.
- Peak in the histogram shows rotational difference between images.
- SIFT rotation parameters appears useful in the absence of scale difference.

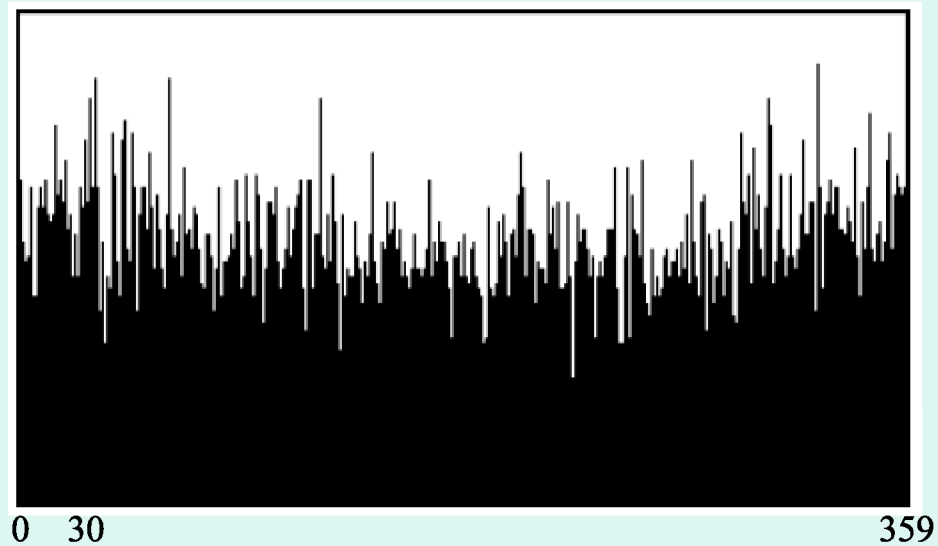


0 30

359 13

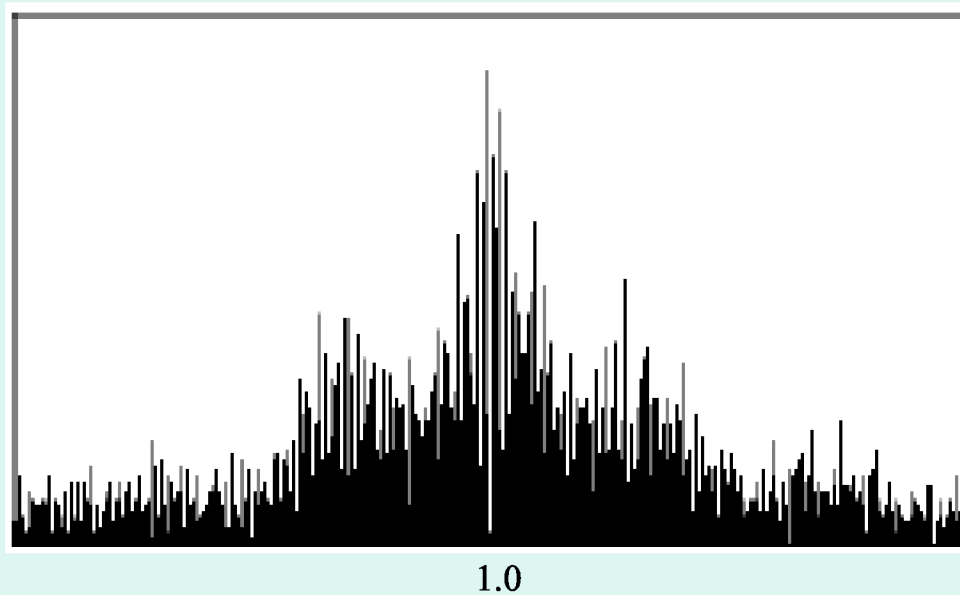
SIFT rotation parameters

- When the images have scale difference, rotation parameters are not very useful.



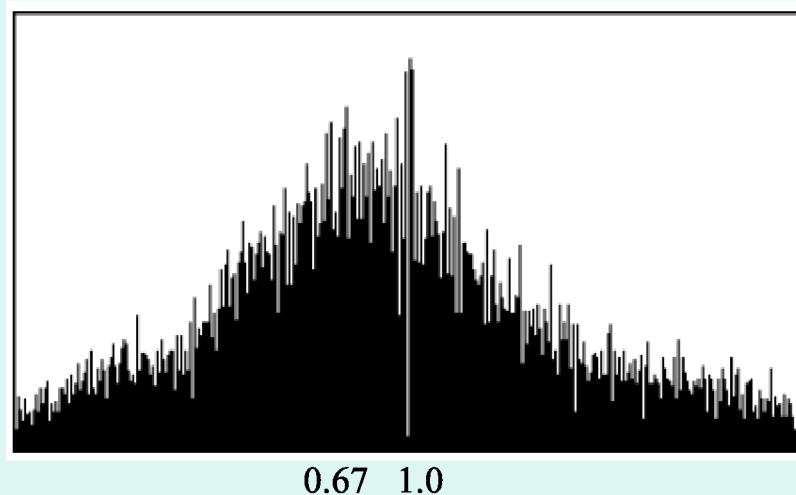
SIFT scale parameters

- When the images are in the same scale, scale parameters are useful. Show below is the histogram of the scale ratio of MST points with the same degree in the original and rotated images.



SIFT scale parameters

- When the images have different scales, scale parameters are not useful. Shown below is the histogram of the scale ratios of original and scaled images.



Related references

- S. Belongie, J. Malik, and J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002. (Shape context)
- S. Lazebnik, C. Schmid, and J. Ponce, Sparse texture representation using affine-invariant neighborhoods, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005. (RIFT)
- D. Lowe, Distinctive image features from scale-invariant keypoints, *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. (SIFT)
- K. Mikolajczyk and C. Schmid, Aperformance evaluation of local descriptors, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp.1615–1630, 2005. (GLOH)
- M. J. Swain and D. H. Ballard, Color indexing, *Int'l J. Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991. (Histogram-based)

The Correspondence Problem

Arthur Goshtasby

Wright State University

Image Registration and Fusion Systems

Correspondence methods

Point pattern matching:

- 1) scene coherence
- 2) clustering
- 3) invariance

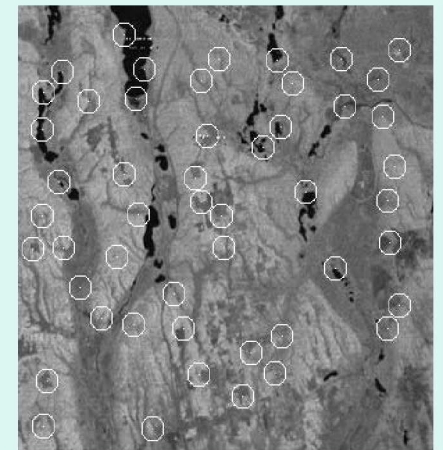
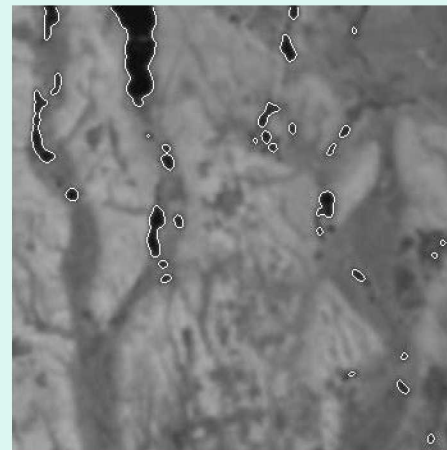
Line matching

Region matching:

- 1) shape matching
- 2) relaxation labeling
- 3) chamfer matching

Template matching:

- 1) Similarity measures
- 2) Coarse-to-fine methods

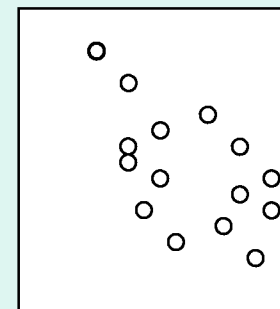
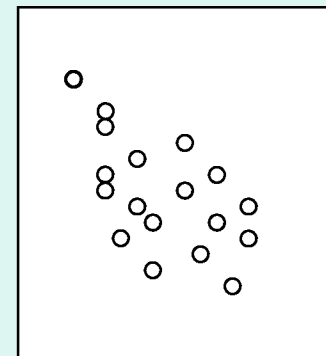


Point pattern matching

Problem: Given two sets of points, determine the correspondence between them.

- Information about only the locations of the points is available.
- The points may contain positional error.
- Some points may exist in only one of the sets.

We will only consider the case where the two point sets are related by the affine transformation.



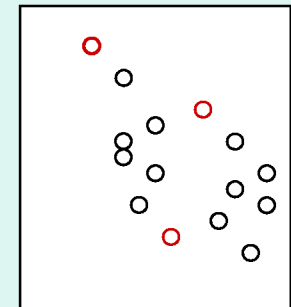
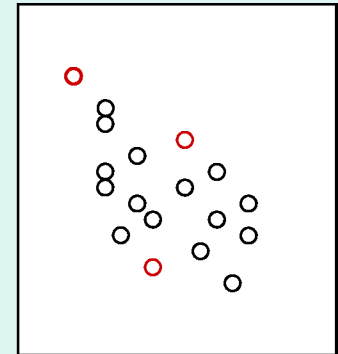
Point pattern matching using scene coherence

If three points in the two sets are aligned, because of the scene coherence, the remaining points in the two sets will also align.

RANSAC Algorithm: Find three corresponding points.

1. **Take 3 points from set 1 and 3 points from set 2 and find the affine transformation that aligns them.**
2. **Count the number of other points in the two sets that also align with the obtained transformation.**
3. **If the count is sufficiently high, stop the process. Otherwise, go to step 1.**

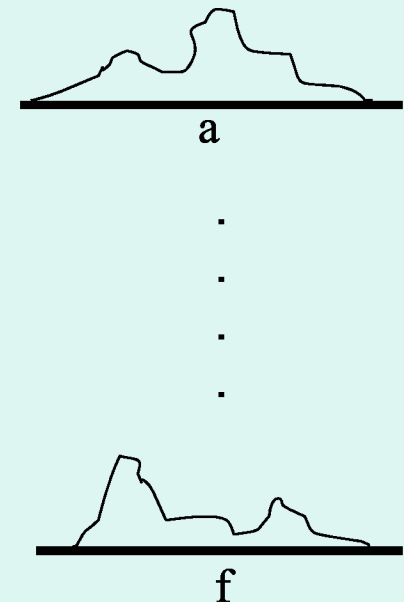
To speed up the search, limit the point combinations to those falling on the convex-hulls or the minimum-spanning trees of the two sets.



Point pattern matching using clustering

Algorithm:

1. Create accumulators $a[] - f[]$ and initialize the entries to 0.
2. From point triples in the two sets calculate parameters $a - f$ of
$$X = ax + by + c,$$
$$Y = dx + ey + f.$$
3. Increment entries $a - f$ of accumulators $a[] - f[]$, respectively, by 1.
4. Repeat Steps 2 and 3 a sufficiently large number of times.
5. Locate the entry with the highest counts in $a[] \dots f[]$. They show parameters $a - f$ of the registration.

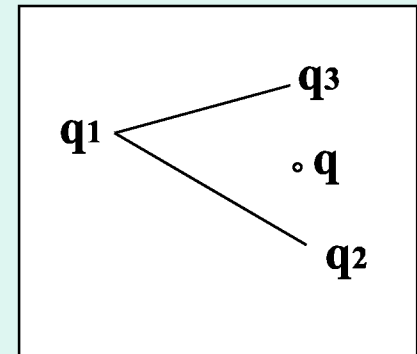
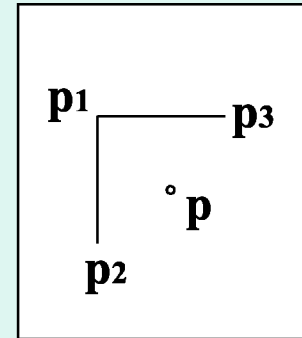


Point pattern matching using affine invariance

If two point sets are related by an affine transformation, knowing three corresponding points in the two sets (p_1, q_1) , (p_2, q_2) , (p_3, q_3) , the relation between corresponding points (p, q) in the sets can be written as

$$p = p_1 + a_1(p_2 - p_1) + a_2(p_3 - p_1)$$

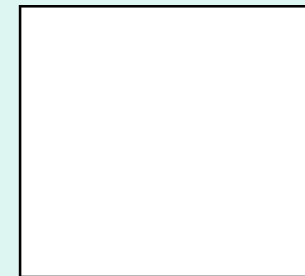
$$q = q_1 + a_1(q_2 - q_1) + a_2(q_3 - q_1)$$



Point pattern matching using affine invariance

Algorithm:

1. Create two 2-D accumulator arrays $\mathbf{H}_1[]$ and $\mathbf{H}_2[]$.
2. Select three points in set 1 and for each additional point in set 1 calculate a_1 and a_2 and increment entry $[a_1, a_2]$ of $\mathbf{H}_1[]$ by 1.
3. Select three points in set 2 and for each additional point in set 2 calculate a_1 and a_2 and increment entry $[a_1, a_2]$ of $\mathbf{H}_2[]$ by 1.
4. Find the similarity between \mathbf{H}_1 and \mathbf{H}_2 . If the similarity is sufficiently high, take the point triples selected in Steps 2 and 3 as corresponding points and stop. Otherwise, go to Step 2.



\mathbf{H}_1



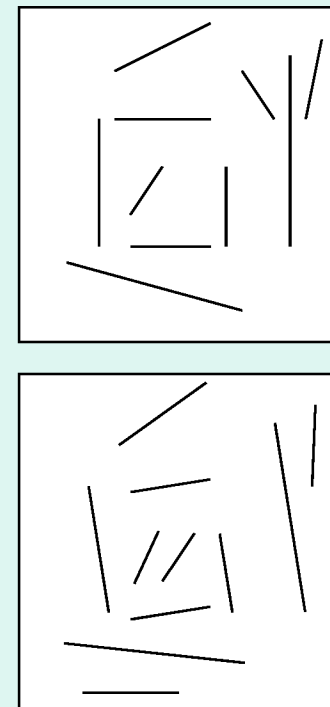
\mathbf{H}_2

Line matching

Assumption: Images are related by a rigid transformation (unknown translation and rotation).

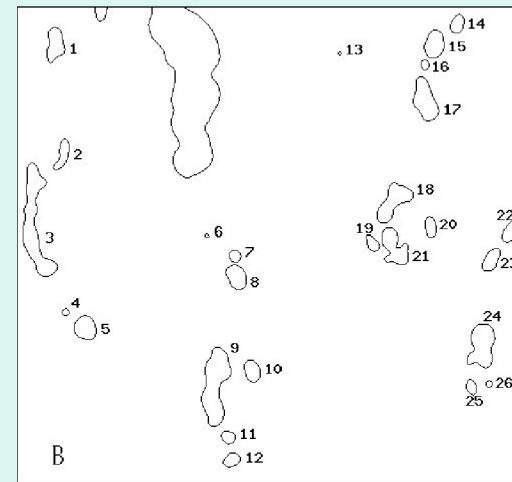
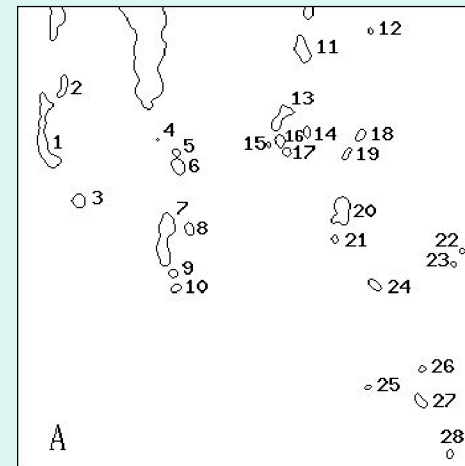
Algorithm:

1. Find the rotational difference between the line sets.
2. Correct the orientation of set 2 with respect to set 1.
3. Find the translational difference between the line sets.



Region matching

- Shape matching
 - Fourier descriptors
 - Invariant moments
 - Shape matrices
 - Relaxation labeling
 - Chamfer matching
 - Distance transform



Fourier descriptors

- Given pixels on the boundary of a region $\{(x_i, y_i): i=0, \dots, N-1\}$, the xy coordinates of the pixels along the boundary can be considered samples from a periodic signal. Letting $z_i = x_i + jy_i$, where $j = \sqrt{-1}$, Fourier descriptor is computed from

$$c_k = \frac{1}{N} \sum_{i=0}^{N-1} z_i \exp(-j2\pi ki / N), \quad k=0, \dots, N-1$$

Invariant moments

- Given boundary pixels $\{(x_i, y_i): i=0, \dots, N-1\}$, the $(p+q)$ th order moment is defined by

$$m_{pq} = \sum_{i=0}^{N-1} x_i^p y_i^q f_i$$

The $(p+q)$ th order central moment of the boundary is defined by

$$u_{pq} = \sum_{i=0}^{N-1} (x_i - \bar{x})^p (y_i - \bar{y})^q f_i$$

- Invariant moments:

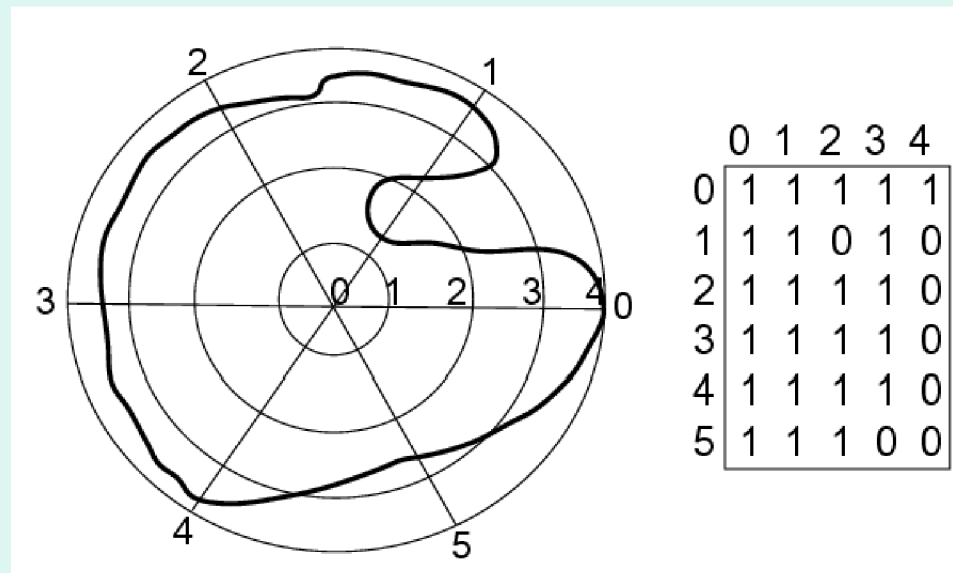
$$a_1 = u_{20} + u_{02}$$

....

$$a_7 = (3u_{21} - u_{03}) \dots$$

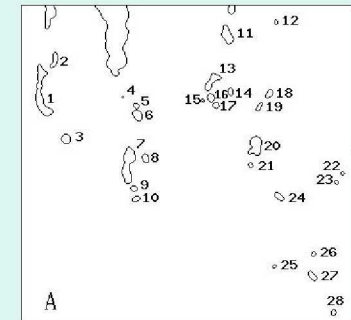
Shape matrix

- Resampling a shape into a representation that is independent of its position, orientation, and scale.

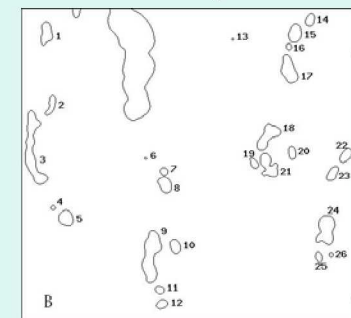


Relaxation labeling

- Denote regions in the reference image by $\{a_i: i=1, \dots, m\}$ and regions in the sensed image by $\{b_i: i=1, \dots, n\}$.
- If a_i 's denote objects and b_i 's denote labels, the correspondence problem becomes that of finding labels for the objects such that a compatibility condition is satisfied.
- Initially assign labels to the objects with probabilities proportional to their similarities.
 $P_i(b_j)$: similarity between regions a_i and b_j .
- Iteratively revise the label probabilities until they converge to either 0 or 1.



Set of objects



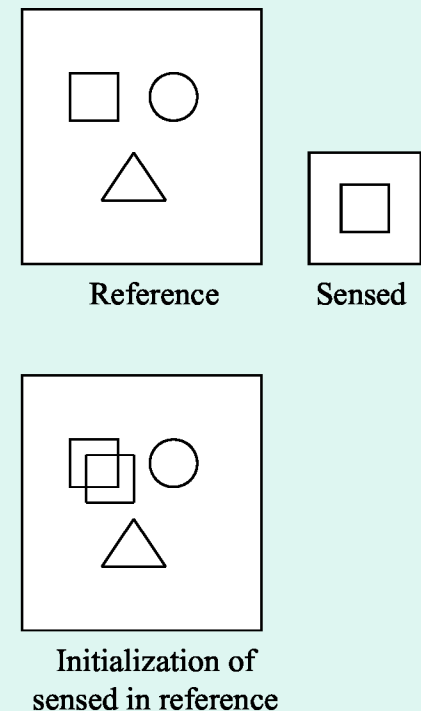
Set of labels

$$P_i^{k+1}(b_j) = \frac{P_i^k(b_j) [1 + q_i^k(b_j)]}{\sum_{j=0}^n P_i^k(b_j) [1 + q_i^k(b_j)]}$$

Chamfer matching

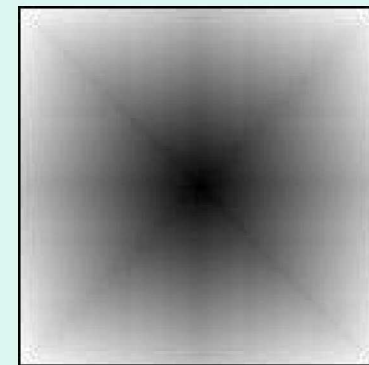
Given two binary images with *translational differences*:

1. Initially position the sensed image within the reference image at (i,j) based on some available information.
2. Determine the similarity between the two images using distances of closest object points.
3. Reposition the sensed image within the reference image at the eight neighbors of (i,j) and determine the image similarities at these eight positions.
4. If highest similarity is obtained when sensed image is at (i,j) , stop. Otherwise, move the sensed image to the neighbor of (i,j) that produces the highest similarity and go to Step 2.

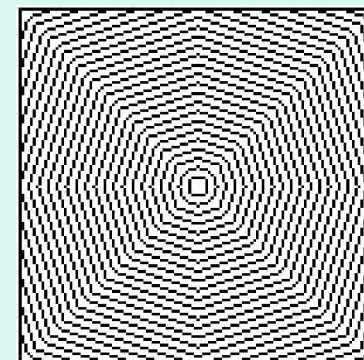


Distance transform

- Given a binary image, assign to a background pixel a value proportional to its distance to the object point closest to it.
- To speed up the computations, integer distances may be used, but note that integer distances involve errors and make distances dependent on the orientation of the image.



Distance transform of a
single point

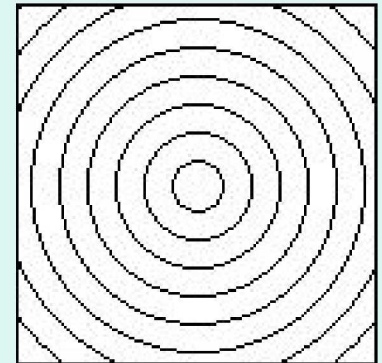
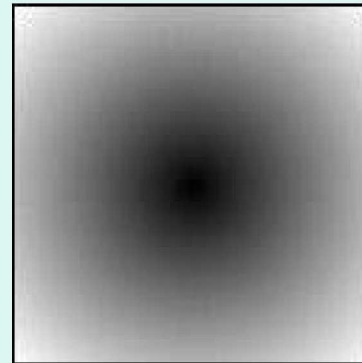


Isovalued distances

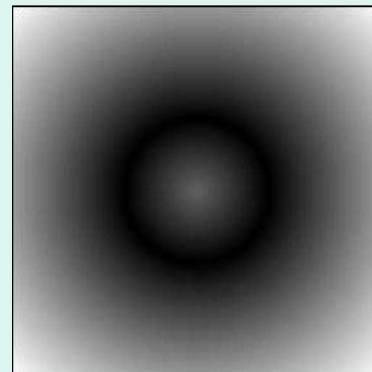
Distance transform

To make distances rotationally invariant, use actual Euclidean distances.

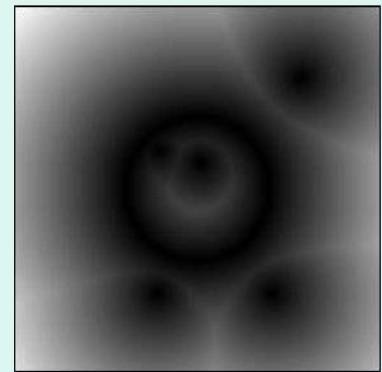
Distance transform in its current form is very sensitive to noise.



Euclidean distance transform of a single point and the iso-valued distances.

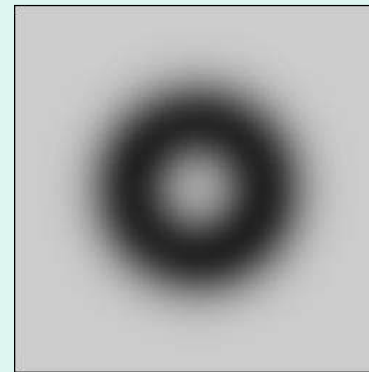


Euclidean distance transform of a circle

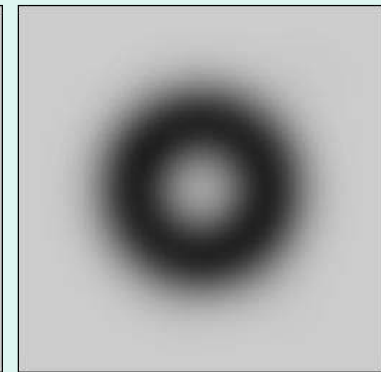


Euclidean distance transform of a circle and 5 noisy points

- Instead of saving a single distance at a pixel, save a weighted sum of distances.

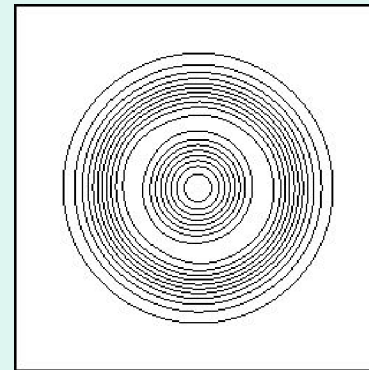


Circle

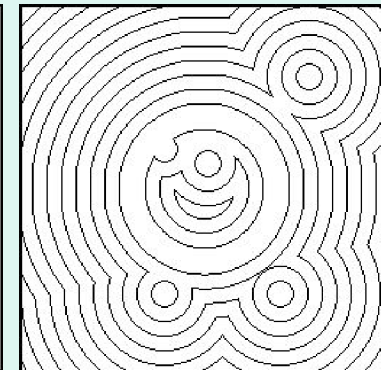


Circle + 5 points

- This can be implemented by smoothing the binary image with a Gaussian and inverting the values.



New DT of circle + 5 points



Traditional DT of circle + 5 points

Template matching

- This is same as chamfer matching except that the images are no longer binary.
- Similarity measures
 - Sum of absolute differences
 - Cross-correlation coefficient
 - Mutual information
- Gaussian weighted templates
- Coarse-to-fine approaches

Similarity measures

- Given two gray scale images with translational differences, shift one image over the other and at each shift position determine the similarity between the two.
- Sum of absolute differences

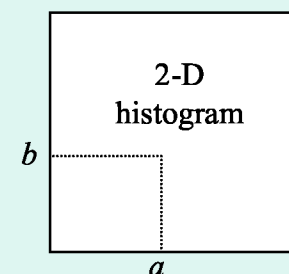
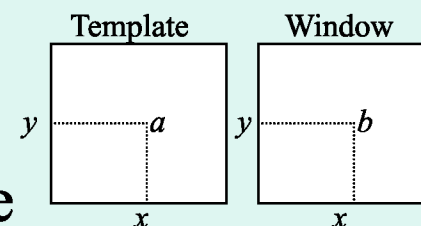
$$D(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |f_t[i, j] - f_w[i + x, j + y]|$$

- Cross-correlation coefficient $(g = f - f_{mean})$

$$S(x, y) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} g_t[i, j] g_w[i + x, j + y]}{\left\{ \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} g_t^2[i, j] \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} g_w^2[i + x, j + y] \right\}^{1/2}}$$

Mutual information

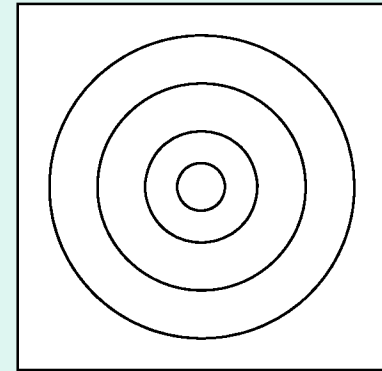
- Create a 2-D histogram with entry $[a, b]$ showing the number of pixels in the template with intensity a that align with intensity b in the window.
- Divide the counts by the number of pixels in template to obtain joint probabilities.
- Then, compute:



$$I(t, w) = \sum_{a=0}^{255} \sum_{b=0}^{255} P_{tw}(a, b) \log \frac{P_{tw}(a, b)}{P_t(a)P_w(b)}$$

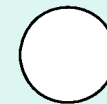
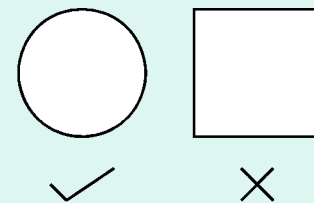
Gaussian weighted templates

- Instead of treating all pixels in a template similarly, give higher weights to pixels that are closer to the template center.
- This makes the process less dependent on image orientation when using rectangular templates.
- It also reduces the effect of geometric difference between images.

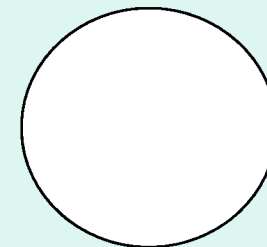


On template shape and size

- Always take circular templates. This will make the process less dependent on image orientation.
- Set template size proportional to the information content in the template. A smaller template size is sufficient when the template is highly detailed compared to when it covers a rather homogeneous area.



For a detailed area



For a less detailed area

Coarse-to-fine matching

- **At image level:** Reduce the size of images, find the correspondences, and determine the transformation parameters. Use the transformation to resample the images at one level higher resolution. Repeat the process until images at the highest resolution are registered.
- **At template level:** Either use smaller templates or cheaper similarity measures to find candidate match positions. Then find the best match position from among the candidates using a larger template and/or a more expensive similarity measure.

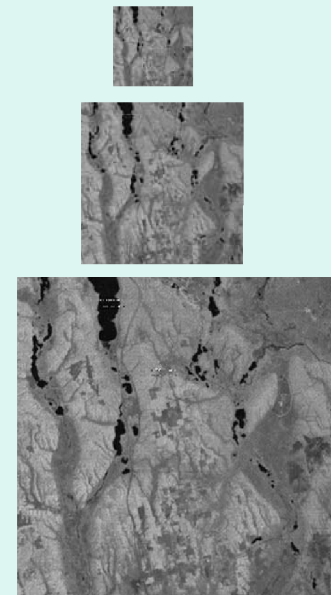
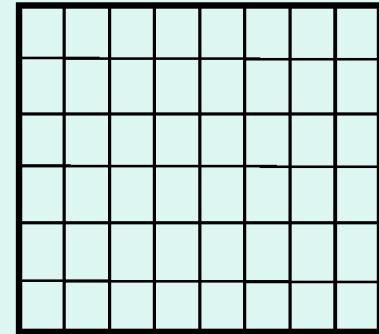


Image pyramid

- **During search:** Use large steps to find possible match positions. Then use finer steps in the neighborhood of the likely matches to refine the final match position.
- **Use partial information:** If a large set of landmarks is given, first use a subset of the landmarks to find approximate registration parameters and then verify the correctness of the registration and refine the parameters using all landmarks.



Feature matching references

1. A. Goshtasby and G. C. Stockman, Point pattern matching using convex-hull edges, *IEEE Transactions on Systems, Man, and Cybernetics*, **15**(5):631–637 (1985).
2. W. J. Rucklidge, Efficiently locating objects using the Hausdorff distance, *Int'l J. Computer Vision*, **24**(3):251–270 (1997).
3. G. Stockman, S. Kopstein, and S. Benett, Matching images to models for registration and object detection via clustering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **4**(3):229–241 (1982).
4. J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, The MIT Press, Cambridge, MA (1992).
5. J. Kittler and J. Illingworth, Relaxation labeling algorithms - A review, *Image and Vision Computing*, **3** (4):206–216 (1985).
6. L. S. Shapiro and J. M. Brady, Feature-based correspondence: An eigenvector approach, *Image and Vision Computing*, **10**(5): 283–288, 1992.
7. A. Goshtasby, S. H. Gage, and J. F. Bartholic, A two-stage cross correlation approach to template matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(3):374–378 (1984).
8. D. Holtkamp and A. Goshtasby, Precision registration and mosaicking of multicamera images, *IEEE Trans. Geoscience and Remote Sensing*, **47**(10):3446–3455, 2009.

Transformation Functions

Arthur Goshtasby

Wright State University

Image Registration and Fusion Systems

Problem description

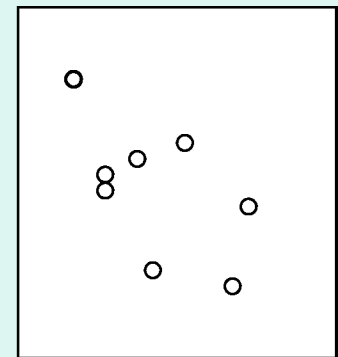
Given the coordinates of a set of corresponding points in the images:

$$\{(x_i, y_i) (X_i, Y_i): i=1, \dots, N\}$$

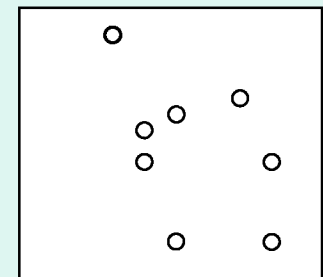
we want to determine function $f(x, y)$ with components $f_x(x, y)$ and $f_y(x, y)$ such that

$$X_i = f_x(x_i, y_i),$$

$$Y_i = f_y(x_i, y_i), \quad i = 1, \dots, N.$$



(x, y)



(X, Y)

Approach

- Rearrange the coordinates of corresponding points into two sets of 3-D points:

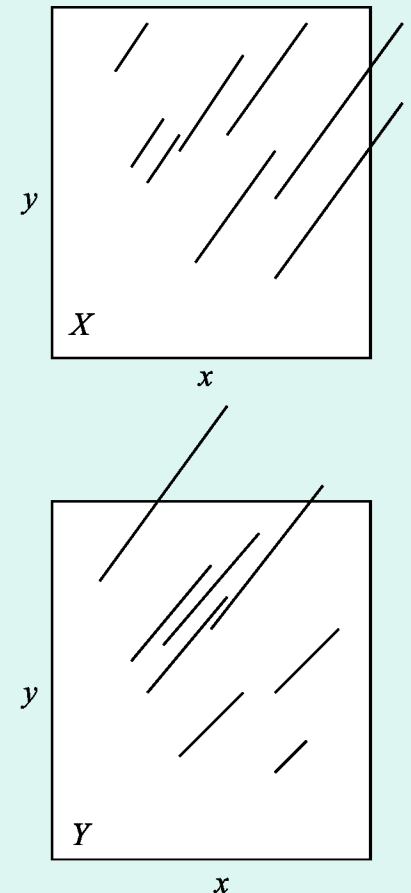
$$\{(x_i, y_i, X_i): i=1, \dots, N\},$$

$$\{(x_i, y_i, Y_i): i=1, \dots, N\},$$

then, f_x and f_y can be considered two single-valued surfaces fitting to two sets of 3-D points.

- We will consider the problem of finding function $f(x, y)$ that approximates/interpolates

$$\{(x_i, y_i, f_i): i=1, \dots, N\}.$$



Transformation functions

- Translation
- Rigid
- Similarity
- Affine
- Projective
- Thin-plate spline
- Multiquadric
- Weighted mean
- Piecewise methods
- Weighted linear

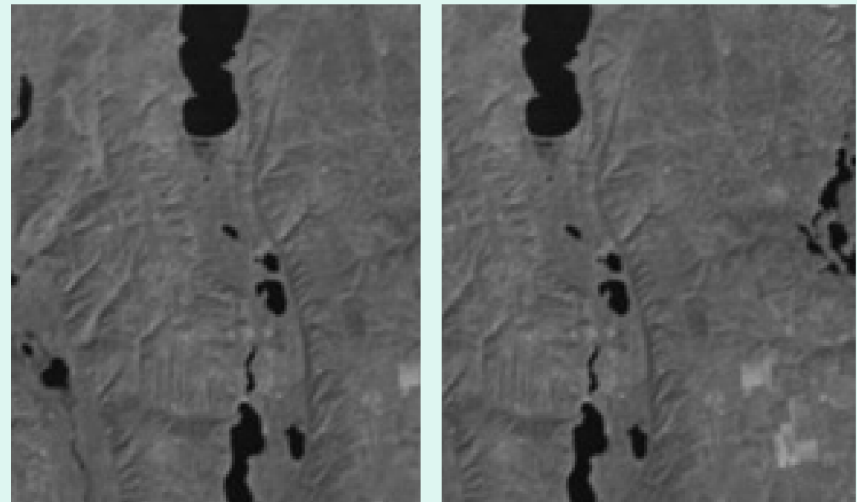
Translation

- Corresponding image points are related by

$$X = x + h$$

$$Y = y + k$$

- One pair of corresponding points is sufficient to determine the registration parameters.

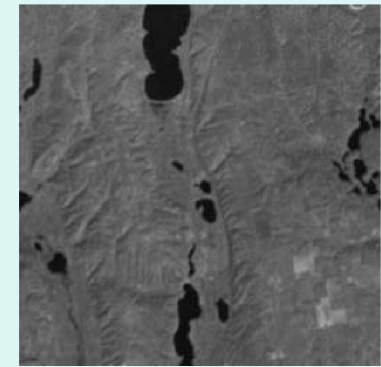


Rigid transformation

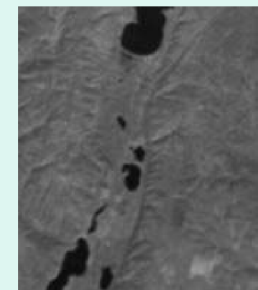
$$X = x \cos(\theta) - y \sin(\theta) + h$$

$$Y = x \sin(\theta) + y \cos(\theta) + k$$

- θ and (h, k) are the rotational and translational differences between the images.
- Knowing minimum of two corresponding points in the images these parameters can be determined.
- This transformation is useful when registering images as rigid bodies.
- Under rigid transformation shape and size are unchanged.



Reference



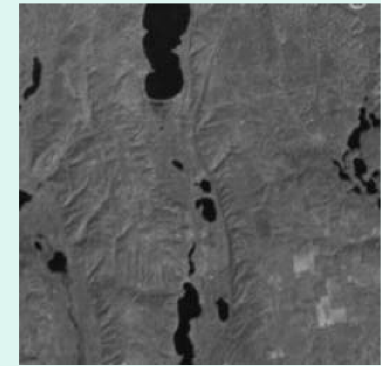
Sensed

Similarity transformation

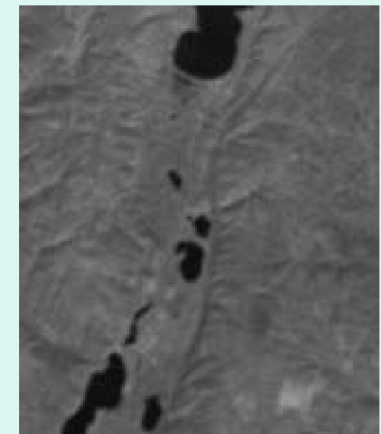
$$X = s x \cos(\theta) - s y \sin(\theta) + h$$

$$Y = s x \sin(\theta) + s y \cos(\theta) + k$$

- s , θ , and (h, k) are the scaling, rotational, and translational differences between the images.
- Minimum two corresponding points in the images are required to determine s , θ , h , and k .
- Angles are preserved under the similarity transformation.
- This transformation is useful when registering distant orthographic images of flat scenes.

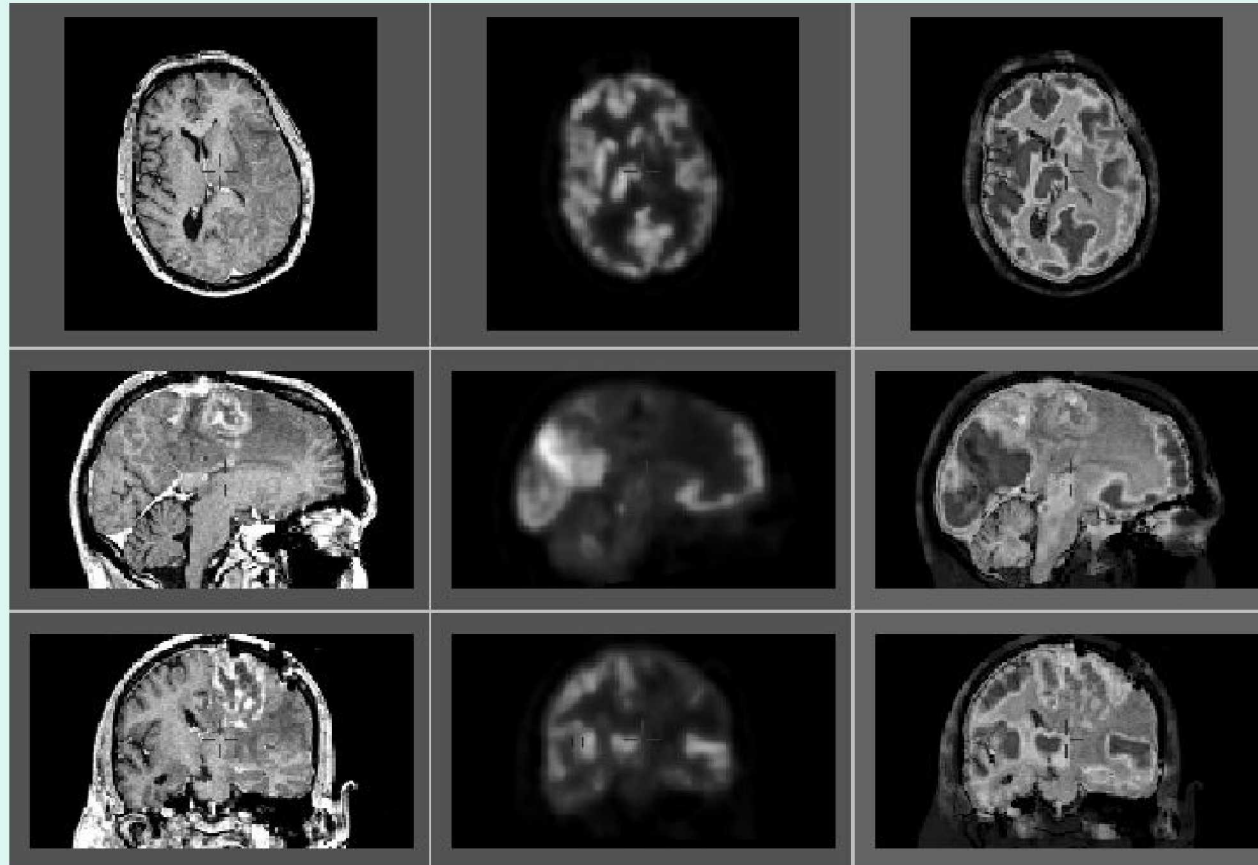


Reference



Sensed

Rigid registration example



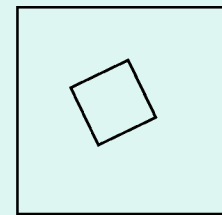
Registration of MR and PET images of the same person.

Affine transformation

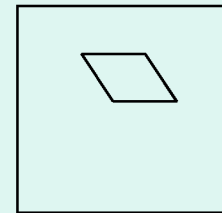
$$X = ax + by + c$$

$$Y = dx + ey + f$$

- This transformation is a combination of shearing and similarity transformations.
- The components of an affine transformation depend on each other.
- Under the affine transformation parallel lines remain parallel.
- When the components are made independent, the transformation becomes a *linear transformation*.
- Knowing the coordinates of three non-collinear corresponding points in the images the six parameters of the transformation can be determined.
- This transformation is useful when registering images taken from a distant platform of a flat scene.



Reference



Sensed

Affine registration example

Lunar
image 1.



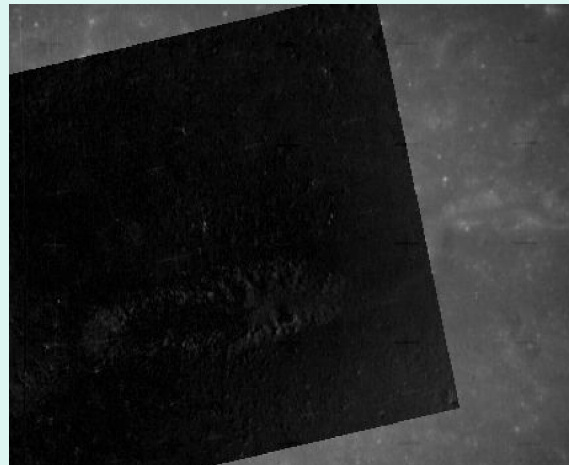
Lunar
image 2.



Registered
lunar
images.



Subtracted
registered
lunar
images.

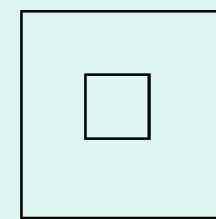


Projective transformation

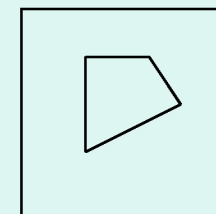
$$X = (ax + by + c) / (dx + ey + 1)$$

$$Y = (fx + gy + h) / (dx + ey + 1)$$

- Knowing the coordinates of four non-collinear corresponding points in the images, parameters $a - h$ can be determined.
- This transformation is useful when registering images obtained from different views of a flat scene.
- Under the projective transformation straight lines remain straight.
- If images are from camera very far from a flat scene, projective transformation can be replaced by the affine transformation when registering the images.



Reference



Sensed

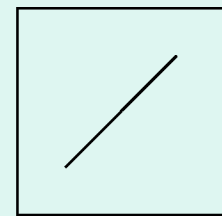
Thin-plate spline (TPS)

$$f(x, y) = A_1 + A_2x + A_3y + \sum_{i=1}^N F_i r_i^2 \ln r_i^2$$

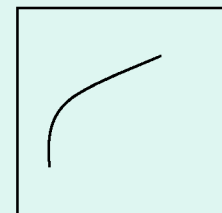
where $r_i^2 = (x - x_i)^2 + (y - y_i)^2 + d^2$

- Parameters A_1, A_2, A_3 , and $F_i: i=1, \dots, N$ are determined using $\{(x_i, y_i, f_i): i=1, \dots, N\}$ and the following three constraints:

$$\begin{aligned}\sum_{i=1}^N F_i &= 0 \\ \sum_{i=1}^N x_i F_i &= 0 \\ \sum_{i=1}^N y_i F_i &= 0\end{aligned}$$



Reference



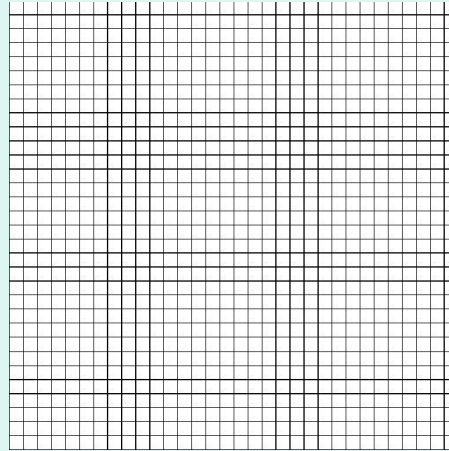
Sensed

Properties of TPS

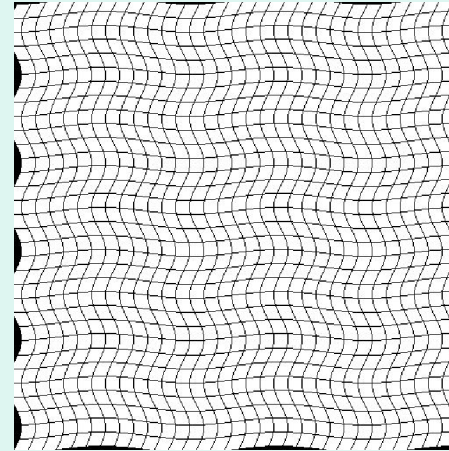
- Logarithmic functions are radially symmetric, so if the control points are not uniformly spaced, large errors may be obtained away from the control points.
- TPS is useful when
 - the local geometric difference between images is not large
 - the control points are rather uniformly spaced
 - the density of the control points does not change
 - the number of corresponding control points is not very high.

Examples

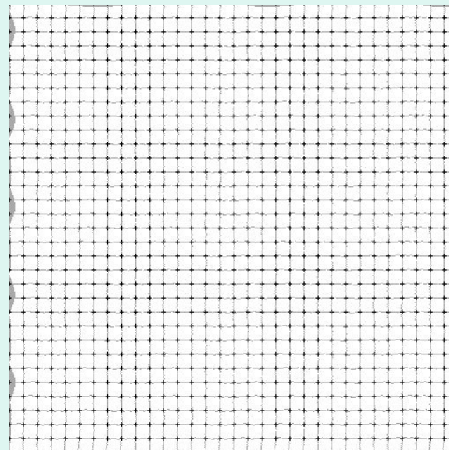
Reference
image.



Sensed
image.

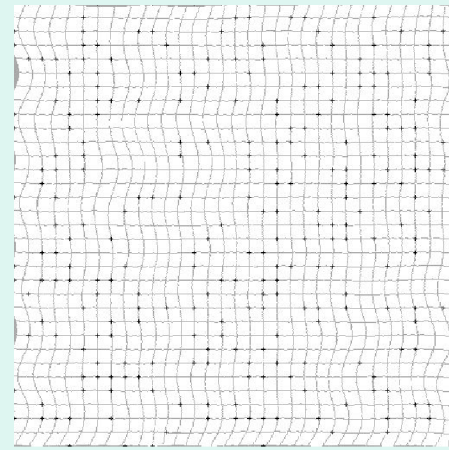


Registered
using all
grid points.



MAX: 0.0; RMS: 0.0

Registered
using
random
grid points.



MAX: 14.1; RMS: 3.5

Multiquadric

Radial basis functions:

$$f(x, y) = \sum_{i=1}^N F_i R_i(x, y)$$

When multiquadrics:

$$R_i(x, y) = [(x - x_i)^2 + (y - y_i)^2 + d^2]^{1/2}$$

When inverse multiquadrics:

$$R_i(x, y) = [(x - x_i)^2 + (y - y_i)^2 + d^2]^{-1/2}$$

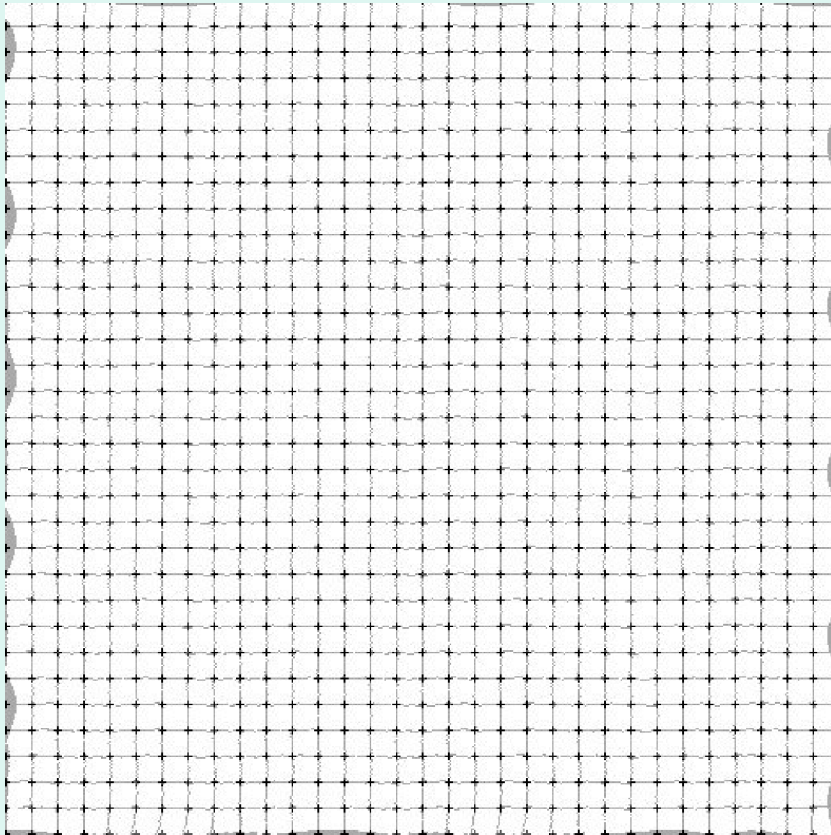
Note that when Gaussian:

$$R_i(x, y) = \exp\left\{-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma^2}\right\}$$

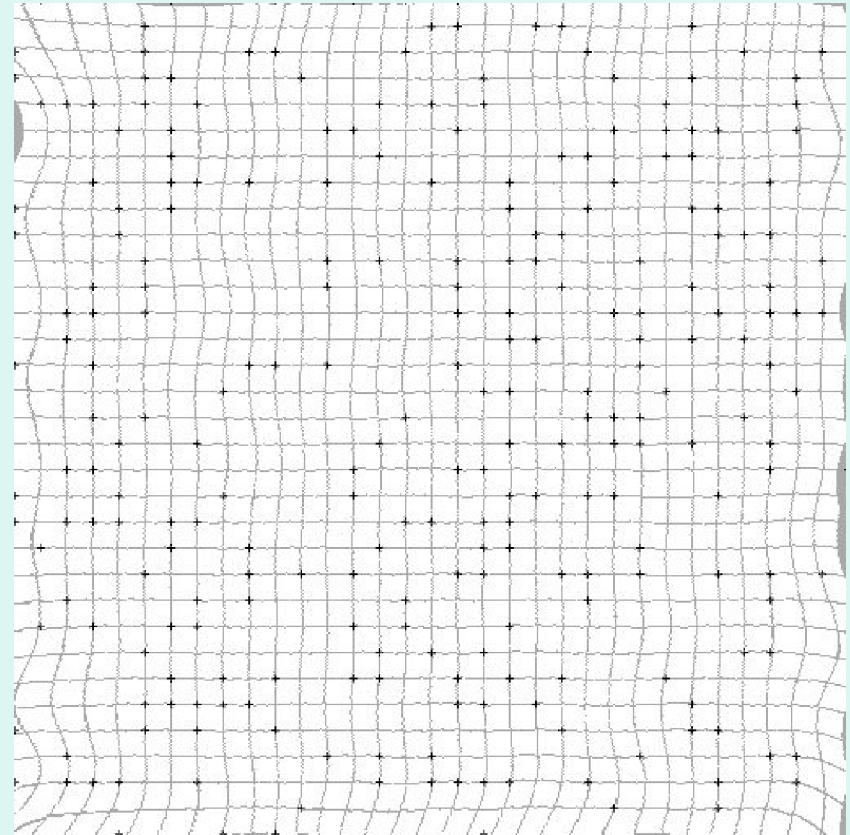
Multiquadric basis functions are also radially symmetric, therefore, their properties are similar to those of TPS.

Examples

MAX: 0.0; RMS: 0.0



MAX: 23.1; RMS: 2.9



Registration using MQ basis functions.

Weighted mean methods

$$f(x, y) = \sum_{i=1}^N f_i b_i(x, y)$$

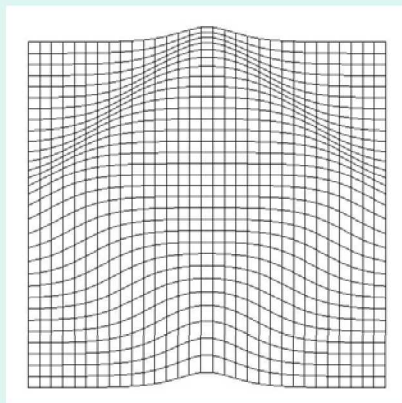
where $b_i(x, y) = R_i(x, y) / \sum_{i=1}^N R_i(x, y)$

and $R_i(x, y)$ is a monotonically decreasing radial function.

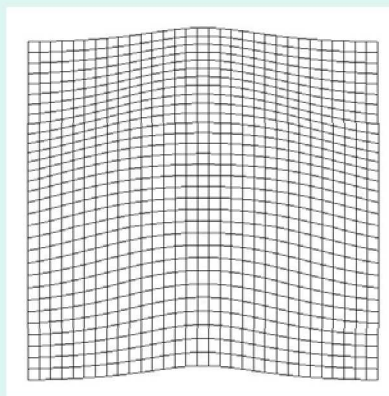
- This is an approximation method; therefore, there is no need to solve a system of equations.

Problem with weighted mean

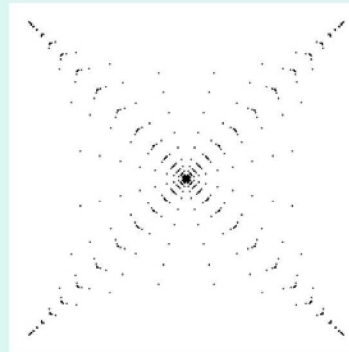
- When the weight functions are narrow, the functions produce flat spots at and in the neighborhood of the data points.
- Flat spots imply that many points in the sensed image map to the same point in the reference image, resulting in large registration errors.
- Consider points: $(0,0,0)$; $(0,1,0)$; $(1,1,0)$; $(1,0,0)$; $(0.5, 0.5, 0.5)$



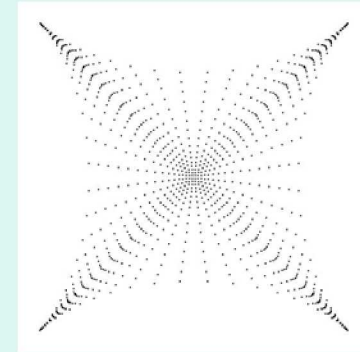
Narrow width



Wider width



Density of points
(narrow width)



Density of points
(wider width)

Remedy

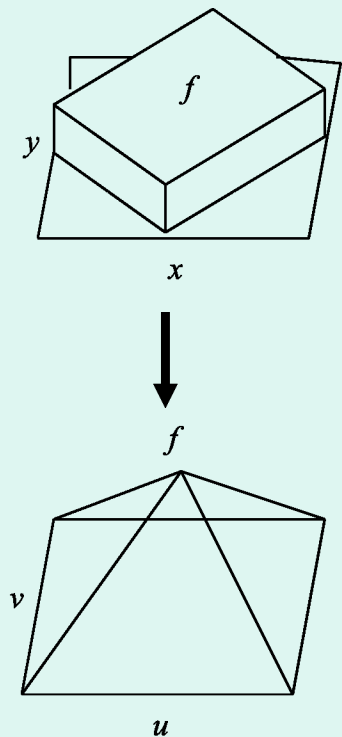
- To produce a uniform density, use parametric surfaces as components of the transformation.

$$X = f_1(u, v) \quad (1)$$

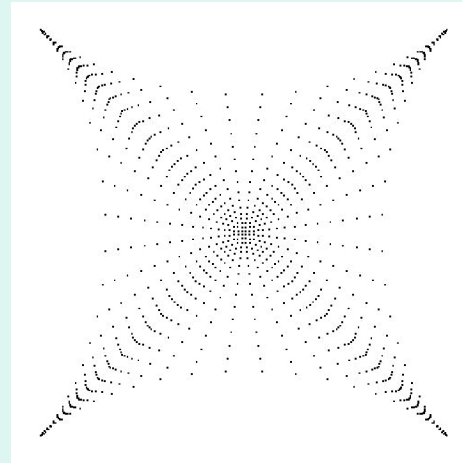
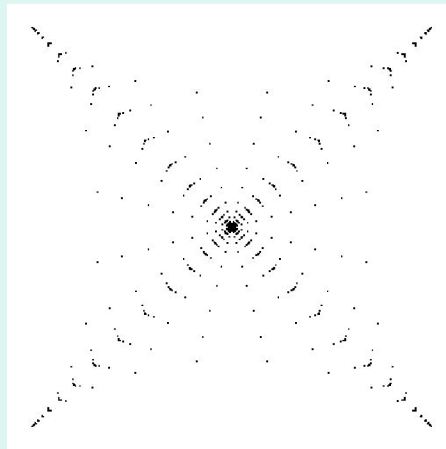
$$x = f_2(u, v) \quad (2)$$

$$y = f_3(u, v) \quad (3)$$

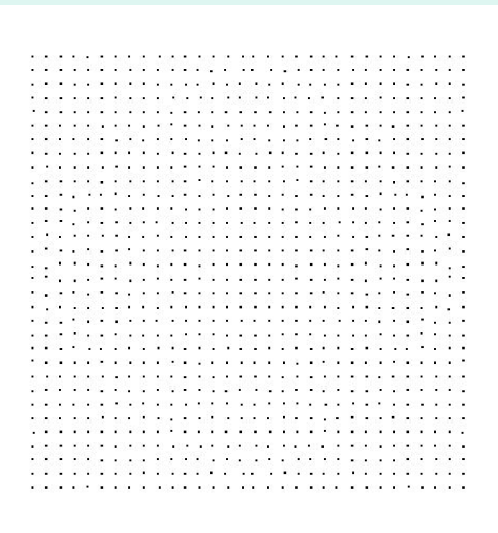
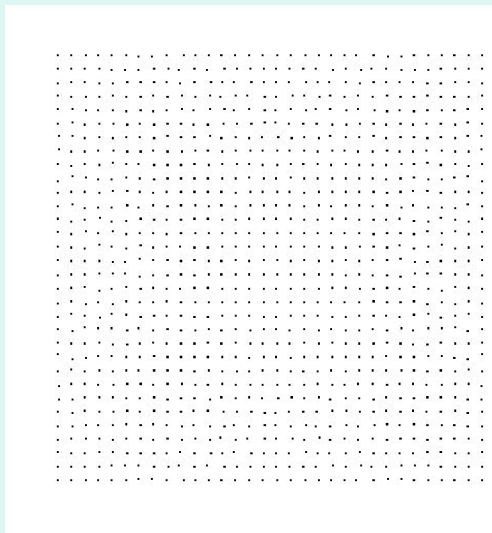
- For a given (x, y) , find corresponding (u, v) from (2) and (3). Then, use (u, v) in (1) to find X .



Resampled
using
single-valued
surfaces.

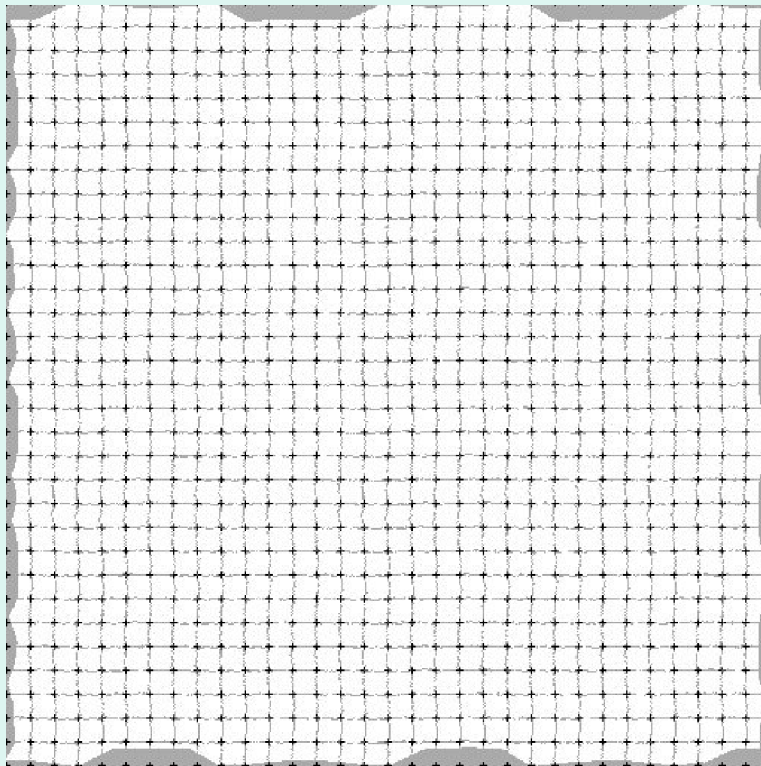


Resampled
using
parametric
surfaces.

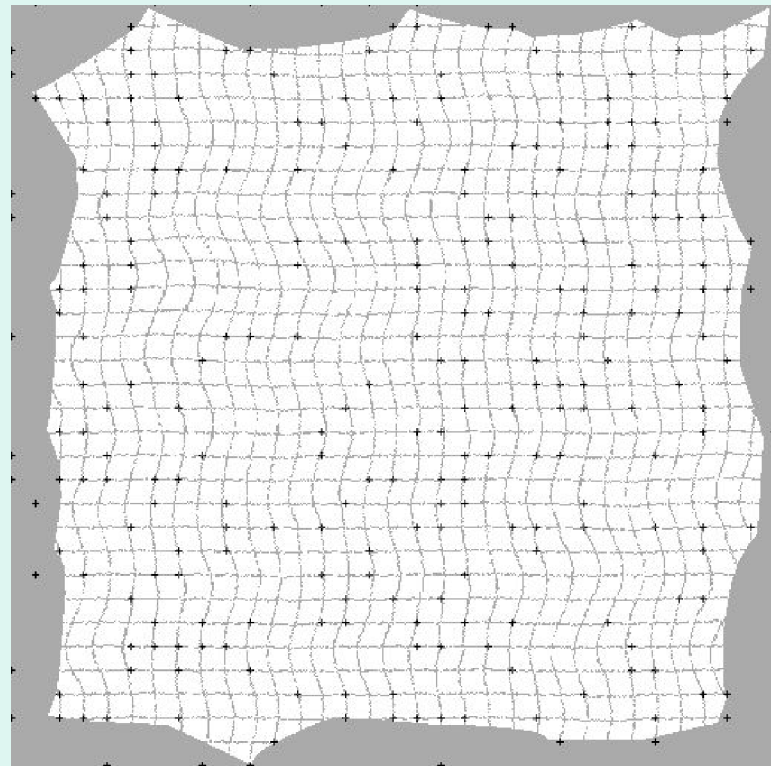


Examples

MAX: 1.4; RMS: 0.6



MAX: 9.0; RMS: 1.3

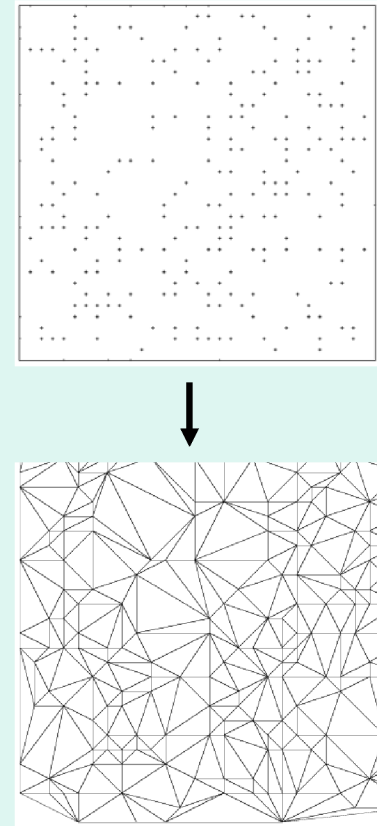


Registered using the weighted mean method.

Piecewise linear and piecewise cubic functions

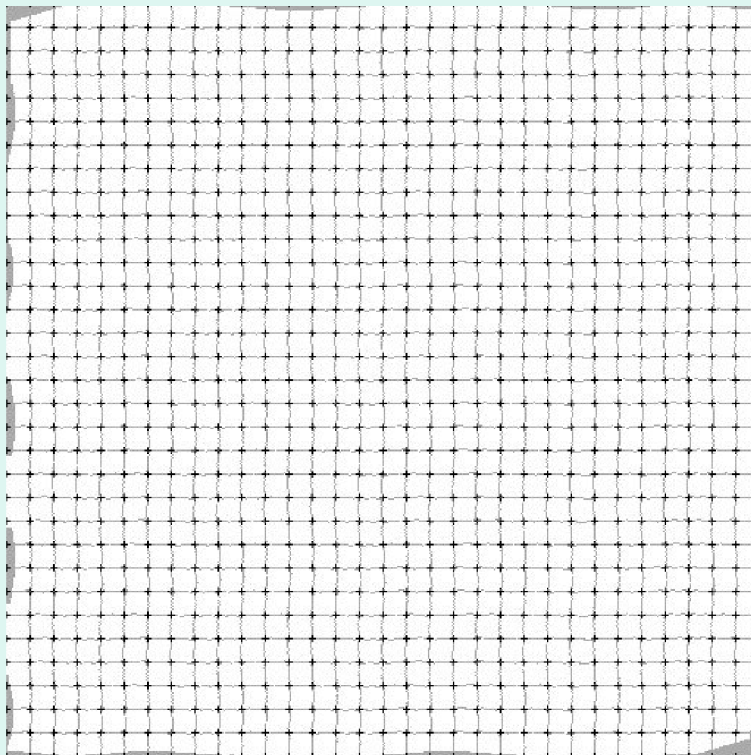
Algorithm:

1. Triangulate the points in the reference image.
2. From the point correspondences, find corresponding triangles in the sensed image.
3. Map triangular regions in the sensed image one by one to the corresponding triangular regions in the sensed image using linear or cubic functions.

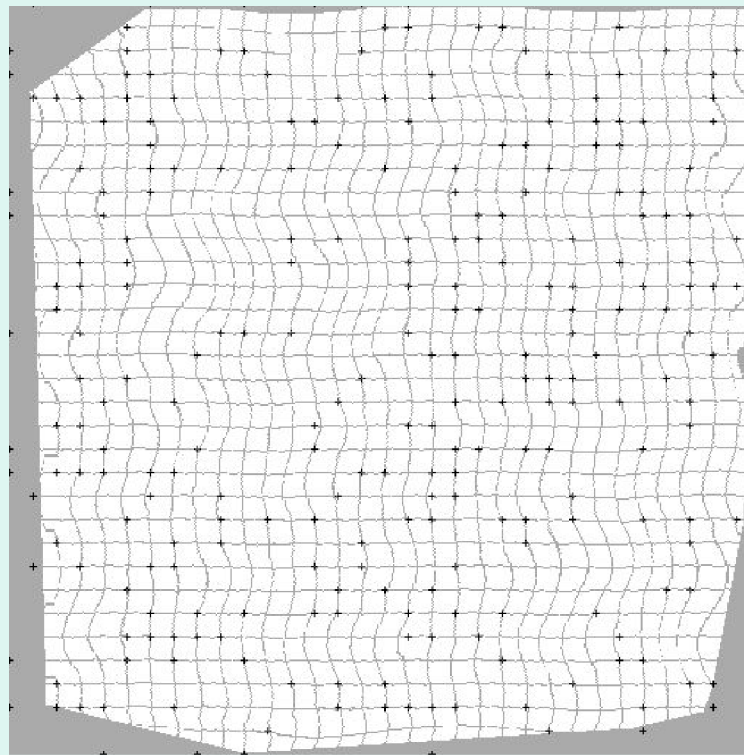


Examples

MAX: 0.0; RMS: 0.0



MAX: 16.1; RMS: 4.0



Registered using piecewise linear functions.

Weighted linear method

- Note that weighted mean is a weighted sum of constants.
- Instead of using a weighted sum of constants, use a weighted sum of linear functions, each representing the geometric difference between the images locally.

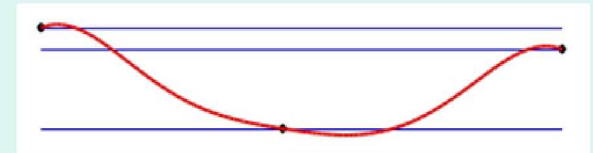
$$f(x, y) = \sum_{i=1}^N f_i(x, y) b_i(x, y)$$

where $f_i(x, y) = a_i x + b_i y + c_i$

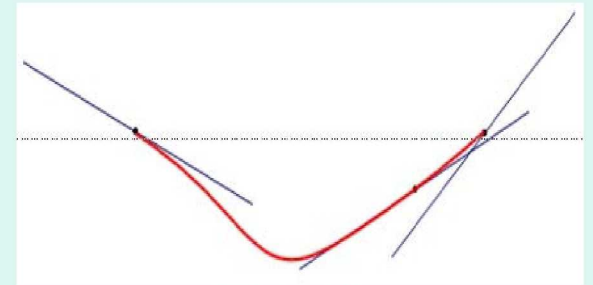
- The local functions at (x_i, y_i) is determined by fitting a plane to (x_i, y_i, f_i) and at least two other points in vicinity of (x_i, y_i) .

Properties of weighted linear functions

- The weighted mean method produces horizontal spots because a surface is obtained from a weighted sum of constants (horizontal planes).
- In weighted linear, since the planes can have any orientation, horizontal spots are not obtained. This implies parametric surfaces are not needed to find the components of a transformation.
- If rational weights are used, because the weights stretch toward the gaps, the functions can adjust themselves to the irregular spacing of the points.
- If the width of the weight functions are set proportional to the local density of points, the functions can adjust themselves to the local density of the points also.



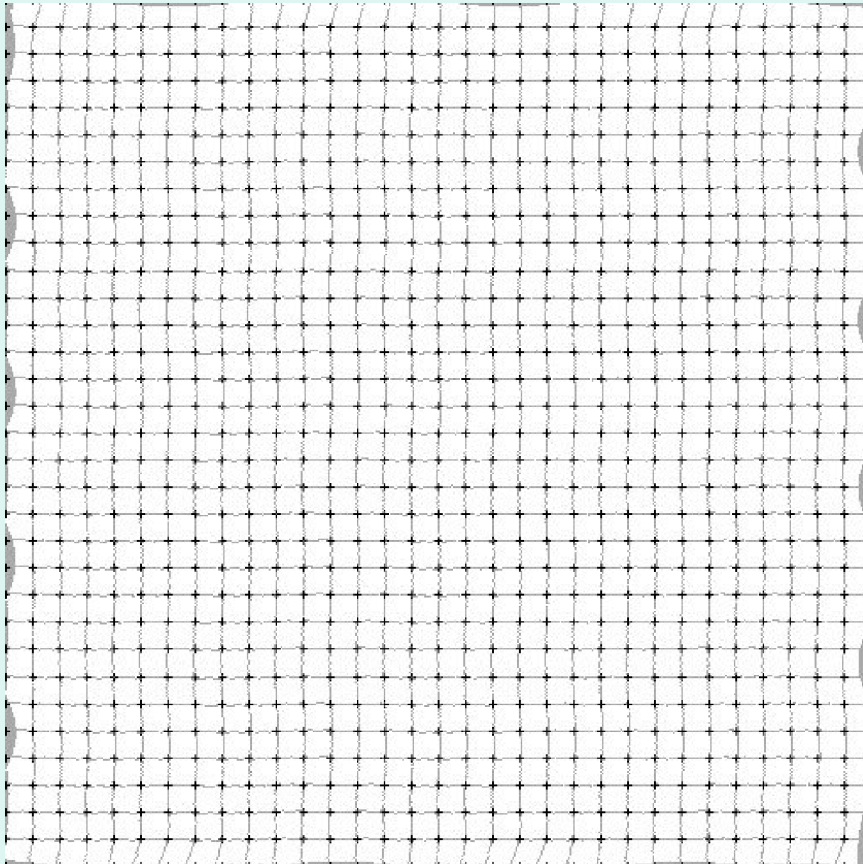
Weighted mean



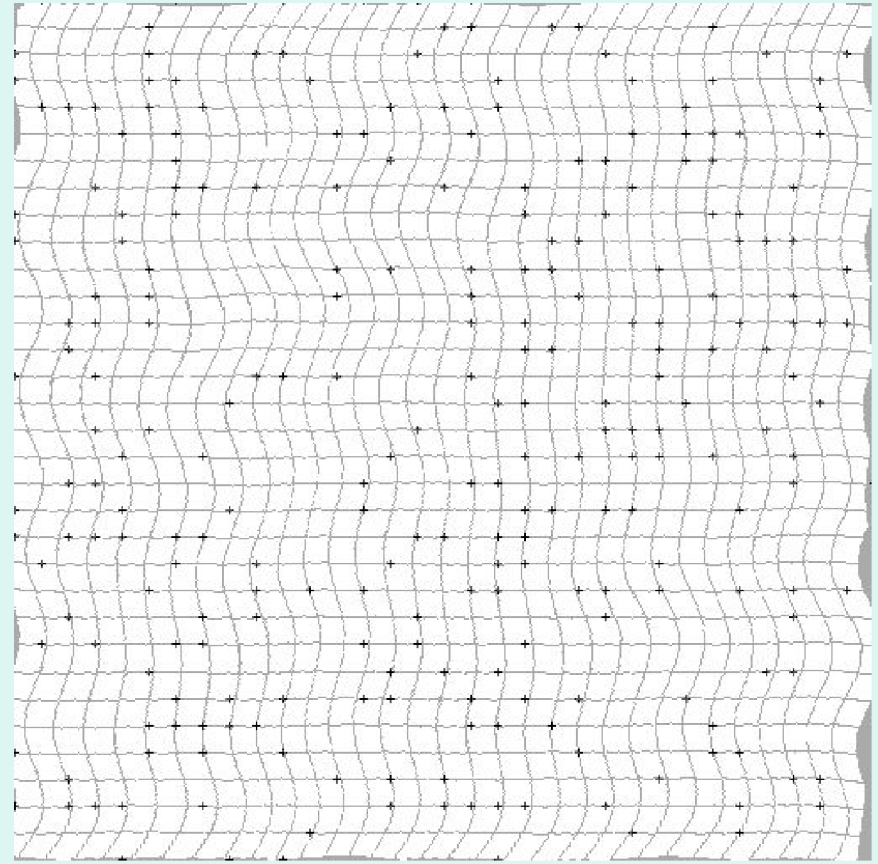
Weighted linear

Examples

MAX: 3.6; RMS: 0.7



MAX: 16.1; RMS: 4.0



Registered using weighted linear functions with rational Gaussian weights.

Computational complexity

Type of Transformation	Computational Complexity
Similarity	$O(N) + O(n^2)$
Affine and Projective	$O(N) + O(n^2)$
Thin-Plate Splines (TPS)	$O(N^2) + O(n^2)$
Multiquadrics (MQ)	$O(N^2) + O(n^2)$
Piecewise Methods	$O(N \log N) + O(n^2)$
Weighted Mean	$O(n^2 N)$
Weighted Linear	$O(n^2 N)$

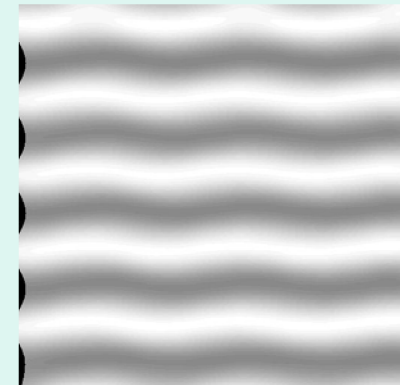
What do transformation functions tell us about the images?

- They contain information about geometric difference between images.

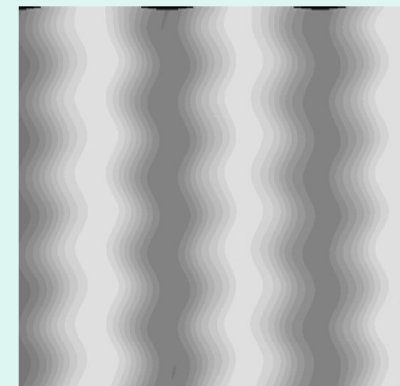
$$f_x(x,y) = x - 8\sin(y/16)$$

$$f_y(x,y) = y + 4\cos(x/32)$$

- They predict the geometry of the scene.



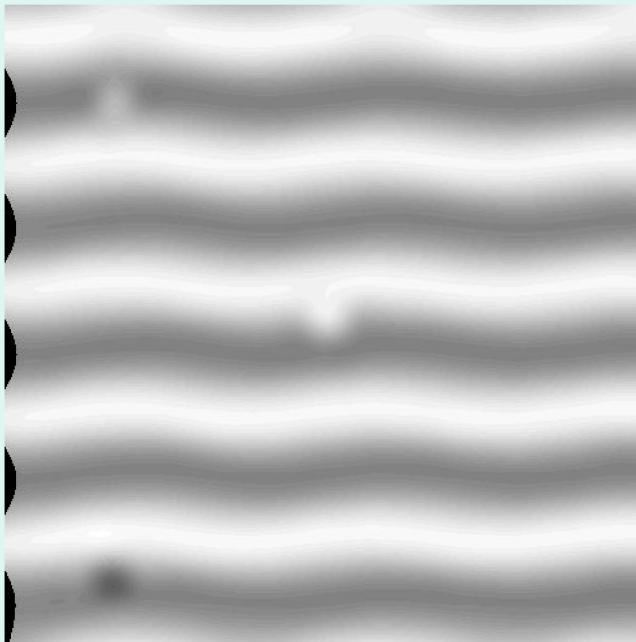
Plot of $f_x(x,y) - x$.



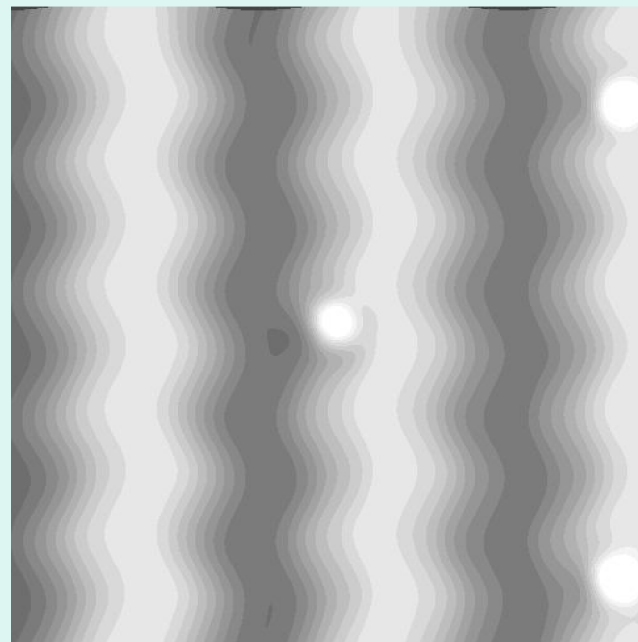
Plot of $f_y(x,y) - y$.

Detecting the mismatches

- Transformation functions contain information about the mismatches.



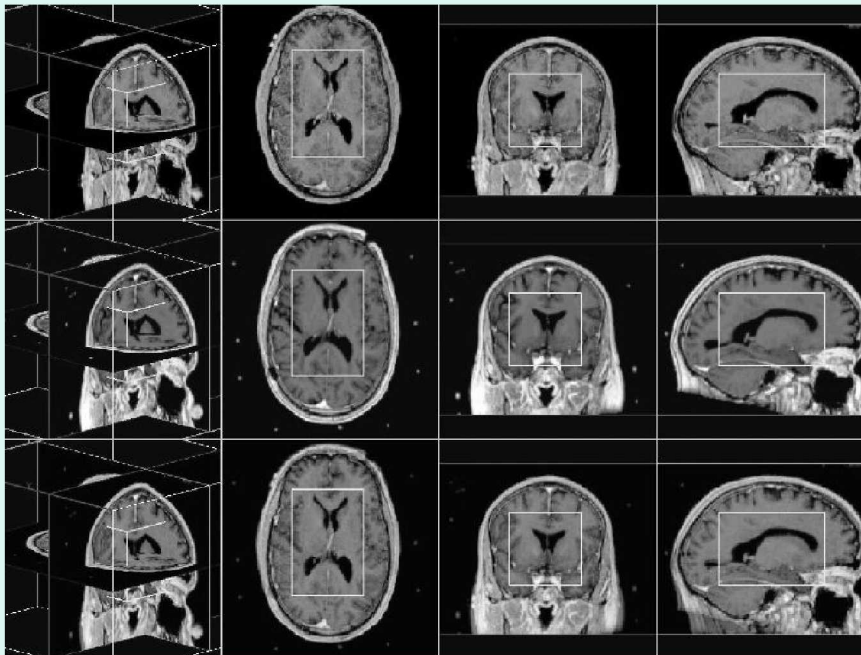
Plot of $f_x(x,y) - x$, using 5 incorrect correspondences



Plot of $f_y(x,y) - y$, using 5 incorrect correspondences

Generating image flow

- Transformation functions can be used to generate flow diagrams, depicting the local motion or deformation of the scene from one image to the next.



Volumetric registration

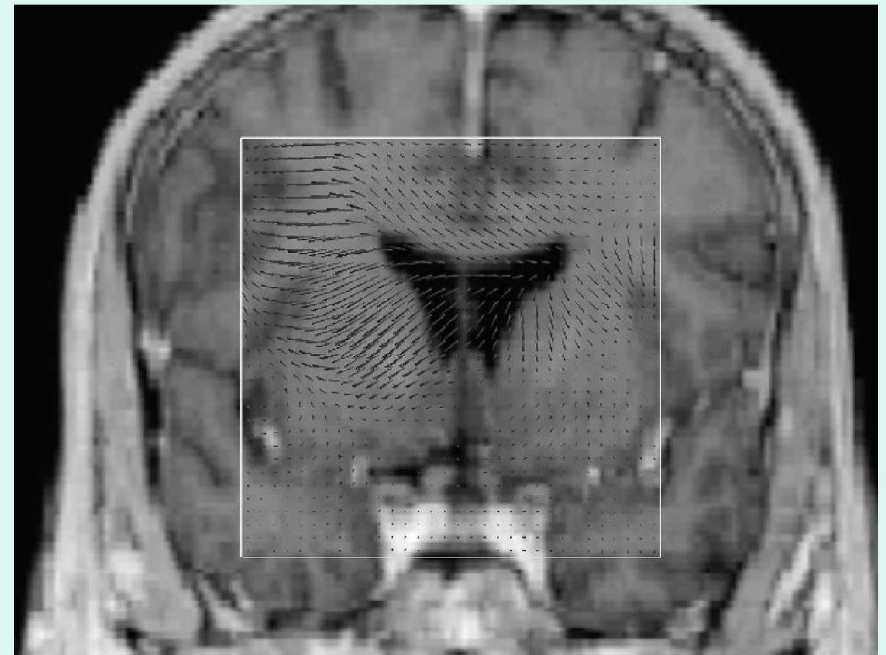


Image flow

References on transformation functions

1. A. Goshtasby, Registration of image with geometric distortions, *IEEE Trans. Geoscience and Remote Sensing*, **26**(1):60–64 (1988).
2. F. L. Bookstein, Principal warps: Thin-plate splines and the decomposition of deformations, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(6):567–585 (1989).
3. K. Rohr, K., H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn, Landmark-based elastic registration using approximating thin-plate splines, *IEEE Transactions on Medical Imaging*, **20**(6):526–534 (2001).
4. M. Fornefett, K. Rohr, and H. S. Stiehl, Radial basis functions with compact support for elastic registration of medical images, *Image and Vision Computing*, **19**:87–96 (2001).
5. A. Goshtasby, Piecewise linear mapping functions for image registration, *Pattern Recognition*, **19**(6):459–466 (1986).
6. A. Goshtasby, Piecewise cubic mapping functions for image registration, *Pattern Recognition*, **20**(5):525–533 (1987).
7. A. Goshtasby, Image registration by local approximation methods, *Image and Vision Computing*, **6**(4):255–261 (1988).

Image Resampling

Arthur Goshtasby

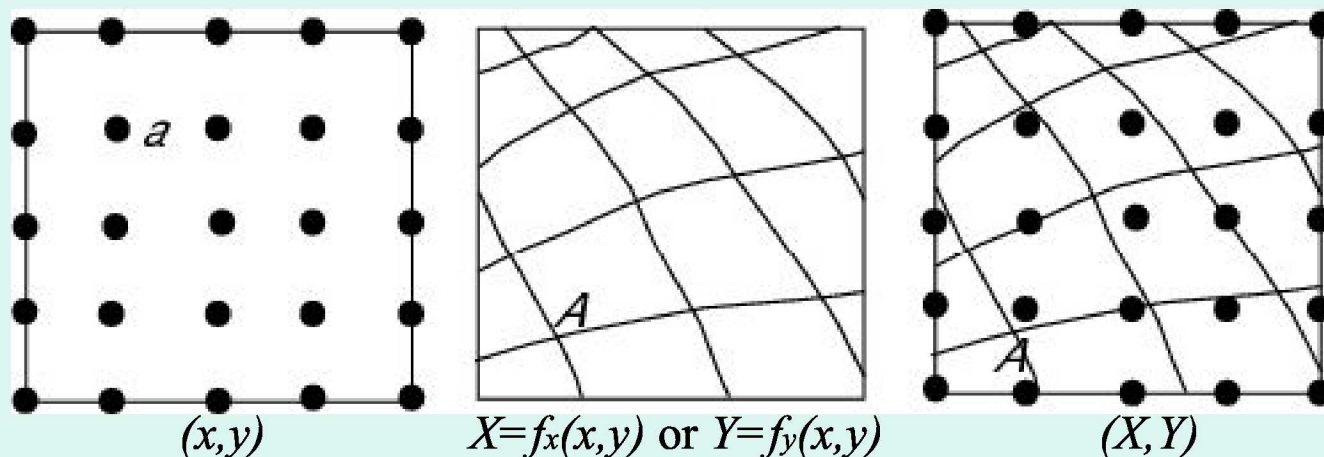
Wright State University

Image Fusion Systems Research

Problem description

Problem: Knowing image intensities at discrete coordinates, determine intensities at arbitrary coordinates.

Approach: Fit a surface to intensities at discrete coordinates and estimate the surface value at desired coordinates.



Resampling methods

- Nearest neighbor
- Bilinear interpolation
- Cubic convolution
- Cubic spline interpolation
- Radially symmetric kernels

Nearest-neighbor resampling

- Set the intensity at (X, Y) to the intensity of the pixel closest to it: $[\text{round}(X), \text{round}(Y)]$.
- This method is very fast, but it produces aliasing effects along edges.

Example:



Original



Resampled

Bilinear interpolation

- Assuming u and v are integer parts of X and Y , respectively, bilinear interpolation is defined by

$$I(X, Y) = W_{u,v}I(u, v) + W_{u+1,v}I(u+1, v) \\ + W_{u,v+1}I(u, v+1) + W_{u+1,v+1}I(u+1, v+1)$$

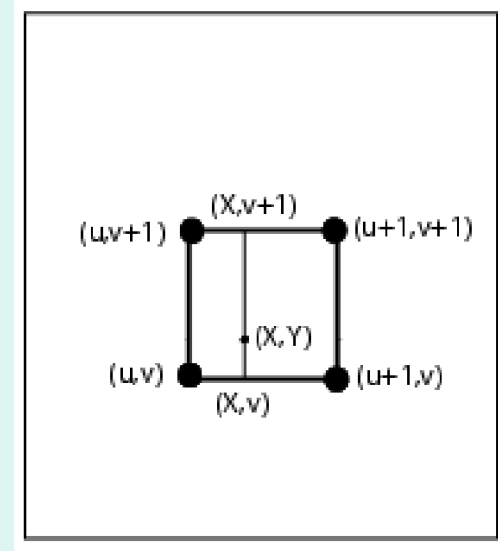
where

$$W_{u,v} = (u+1-X)(v+1-Y)$$

$$W_{u+1,v} = (X-u)(v+1-Y)$$

$$W_{u,v+1} = (u+1-X)(Y-v)$$

$$W_{u+1,v+1} = (X-u)(Y-v)$$



Bilinear example



Original



Resampled

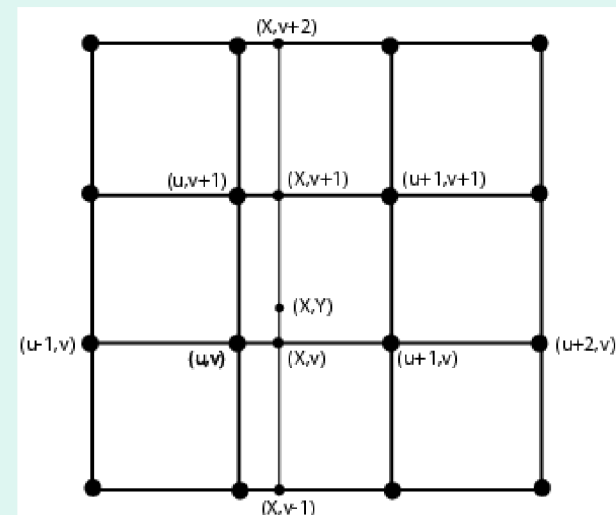
Cubic convolution

- Cubic convolution can be computed row-by-row and then column-by-column.
- Assuming intensities at $u-1$, u , $u+1$, $u+2$ are $I(u-1)$, $I(u)$, $I(u+1)$, $I(u+2)$, intensity at X is estimated from $f(X)=I(u-1)f_{-1}+I(u)f_0+I(u+1)f_1+I(u+2)f_2$

where

$$\begin{aligned} f_{-1} &= -\frac{1}{2}t^3 + t^2 - \frac{1}{2}t, \\ f_0 &= \frac{3}{2}t^3 - \frac{5}{2}t^2 + 1, \\ f_1 &= -\frac{3}{2}t^3 + 2t^2 + \frac{1}{2}t, \\ f_2 &= \frac{1}{2}t^3 - \frac{1}{2}t^2, \end{aligned}$$

and $t = X - u$.



Cubic convolution example



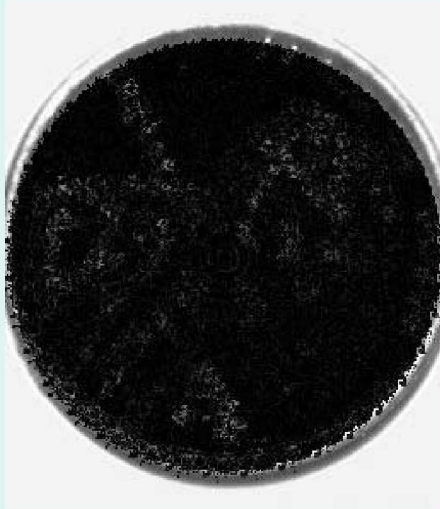
Original



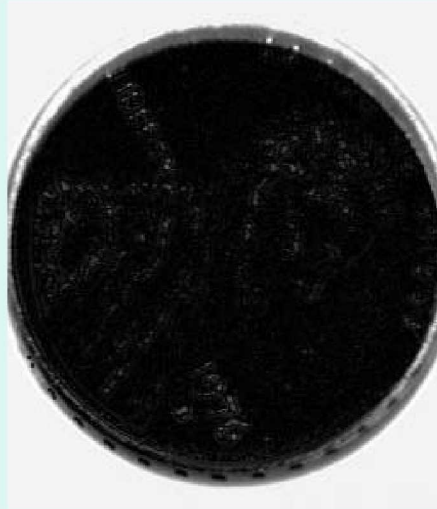
Resampled

Comparison

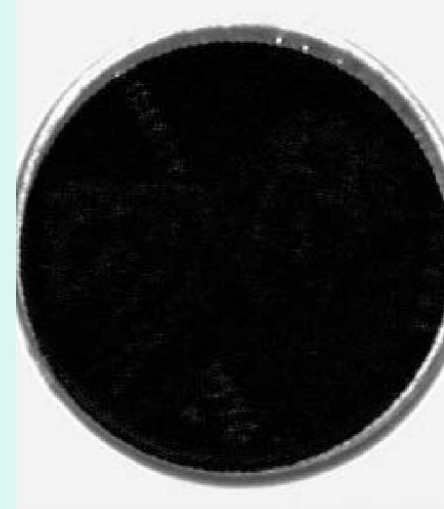
- Suppose an image is rotated with a 10-degree increment 36 times and then subtracted from the original image.
- Only nearest-neighbor preserves original image intensities. Bilinear and cubic convolution change image intensities.



Nearest-neighbor



Bilinear



Cubic convolution

Cubic spline interpolation

- In cubic spline also, computation is carried out row-by-row and then column-by-column.
- Assuming intensities at $i = -1, 0, 1, 2$ are I_{-1}, I_0, I_1, I_2 , the cubic B-spline curve approximating the intensities in the range $0 < u < 1$ is

$$f(u) = \sum_{i=-1}^2 I_i b_i(u)$$

where

$$\begin{aligned}b_{-1}(u) &= (-u^3 + 3u^2 - 3u + 1)/6, \\b_0(u) &= (3u^3 - 6u^2 + 4)/6, \\b_1(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, \\b_2(u) &= u^3/6,\end{aligned}$$

- For the curve to interpolate the intensities, it is required to determine new intensities I'_j s using I_i s.

$$I_i = \sum_{j=-1}^2 I'_j b_j(u_i) \quad i=-1, \dots, 2$$

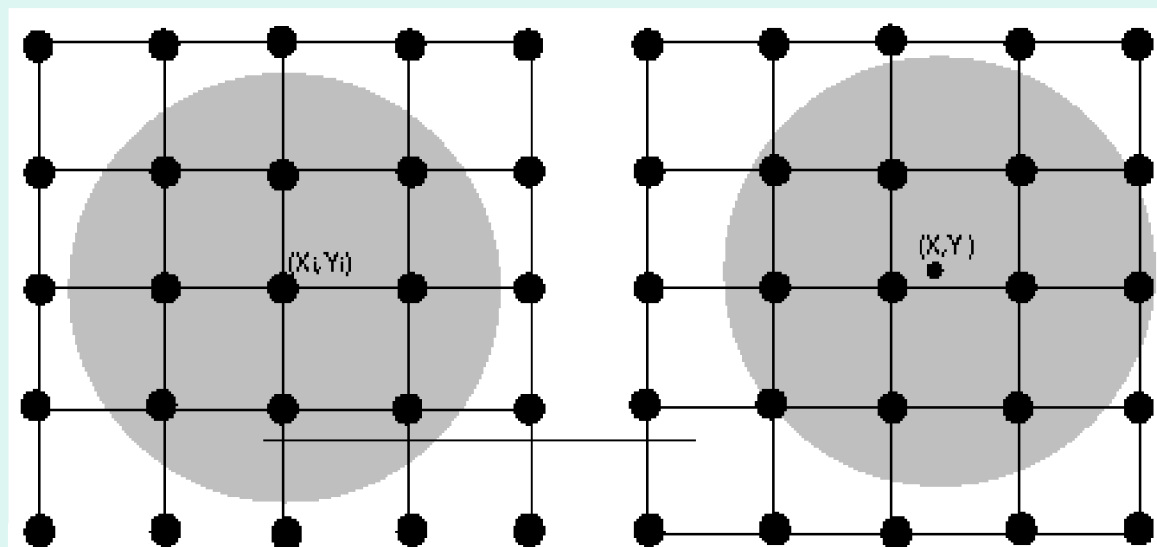
- Instead of solving a very large system of equations, inverse filtering may be used to find the I'_j s.

Radially symmetric kernels

- Cubic convolution and cubic spline methods are not rotationally invariant. To obtain a rotationally invariant resampling, radially symmetric kernels are needed.
- An example of a radially symmetric kernel:

$$f(r_i) = \begin{cases} 1 - 3r_i^2 + 2r_i^3, & 0 \leq r_i \leq 1, \\ 0, & r_i > 1, \end{cases}$$

- The influence of a pixel on interpolating function is radially symmetric.
- The resampled value at a point is obtained from the values of pixels within a circular area centered at the point.



Computational complexity

Type of Resampling	Computational Complexity
Nearest-Neighbor	$O(n^2)$
Bilinear Interpolation	$O(n^2)$
Cubic Convolution	$O(n^2)$
Cubic Spline, Direct Computation	$O(n^4)$
Cubic Spline, Using FFT	$O(n^3 \log n)$
Radial Functions with Local Support	$O(n^4)$
Gaussian, Using FFT	$O(n^3 \log n)$

For an $n \times n$ image.

Resampling references

1. E. G. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoustics, Speech, and Signal Processing*, **29**(6):1153–1160 (1981).
2. H. S. Hou and H. C. Andrews, Cubic splines for image interpolation and digital filtering, *IEEE Trans. Acoustics, Speech, and Signal Processing*, **26**(6):508–517 (1978).
3. L. A. Ferrari, P. V. Sankar, J. Sklansky, and S. Leeman, Efficient two-dimensional filters using B-spline functions, *Computer Vision, Graphics, and Image Processing*, **35**:152–169 (1986).
4. F. Cheng and A. Goshtasby, A parallel B-spline surface fitting algorithm, *ACM Trans. Graphics*, **8**(1):41–50 (1989).
5. A. Goshtasby, F. Cheng, and B. A. Barksy, B-spline curves and surface viewed as digital filters, *Computer Vision, Graphics, and Image Processing*, **52**:264–275 (1990).

Need for Fast and Accurate Image Registration

- Earth Science studies, e.g.:
 - Predicting crop yield
 - Evaluating climate change over multiple scales
 - Locating arable land and water resources
 - Monitoring pollution
 - Understanding the impact of human activity on major Earth ecosystems, etc.
- Global and repetitive measurements from a wide variety of satellite remote sensing systems

Some Examples of Complementary Earth Science Missions

		0.10.40.50.60.71.01.32.03.04.05.06.07.08.09.010.011.012.013.014.015.0																														
Instrument (Spat. Resol.)	Number of Channels	Ultra Violet	Visible				Near-IR			Mid-IR			Thermal-IR																			
AVHRR (D) (1.1 km)	5 Channels				1		2				3											4	5									
TRMM/VIRS (2 km)	5 Channels				1					2					3											4	5					
Landsat4-MSS (80 m)	4 Channels				1	2	3	4																								
Landsat5&7-TM&ETM+ (30 m)	7 Channels			1	2	3	4					5		7											6							
Landsat7-Panchromatic (15m)					1																											
IRS-1 LISS-1 (73m) - LISS-2 (36.5m)	4 Channels			1	2	3	4																									
JERS-1 (Ch1-4:18m; Ch5-8:24m)	8 Channels				1	2	3 &4				5		6	7	8																	
SPOT-HRV Panchromatic (10m)	1 Channel				1																											
Spot-HRV Multispectral (20 m)	3 Channels				1	2		3																								
MODIS (Ch1-2:250 m;3-7:500m;8-36:1km)	36 Channels		3, 8-10	11, 4, 12	1, 13, 14	15	2, 16-19	5	26	6	7				20-25				27	28			29	30			31	32			33-36	
EO/1 ALI-MultiSpectr. ALI-Panchrom. Hyperion LAC	9 Channels (30m) 1 Channel (10m) 220 Channels 256 Channels (250m)		1'	1	2	3	4			5'	5		7																			
IKONOS-Panchromatic (1m)	1 Channel				1																											
IKONOS-MS (4m)	4 Channels (4m)				1	2	3	4																								
ASTER (Ch1-3:15m;4-9:30m;10-14:90m)	14 Channels				1	2	3					4		5-9							10,11	12		13,14								
CZCS (1 km)	6 Channels			1	2	3	4	5																								
SeaWiFS (D) (1.1 km)	8 Channels			1	2	3	4	5	6	7	8																					
TOVS-HIRS2 (D) (15 km)	20 Channels					20									19	18	17 to 13			12	11	10			9			8			7 to 1	
GOES (1 km:1, 4km:2,4&5, 8km:3)	5 Channels				1									2		3							4		5							
METEOSAT (V:2.5km, WV&IR:5km)	3 Channels			Visible										Water Vapor				IR														

Landsat ETM and IKONOS Registration US, Virginia Coast

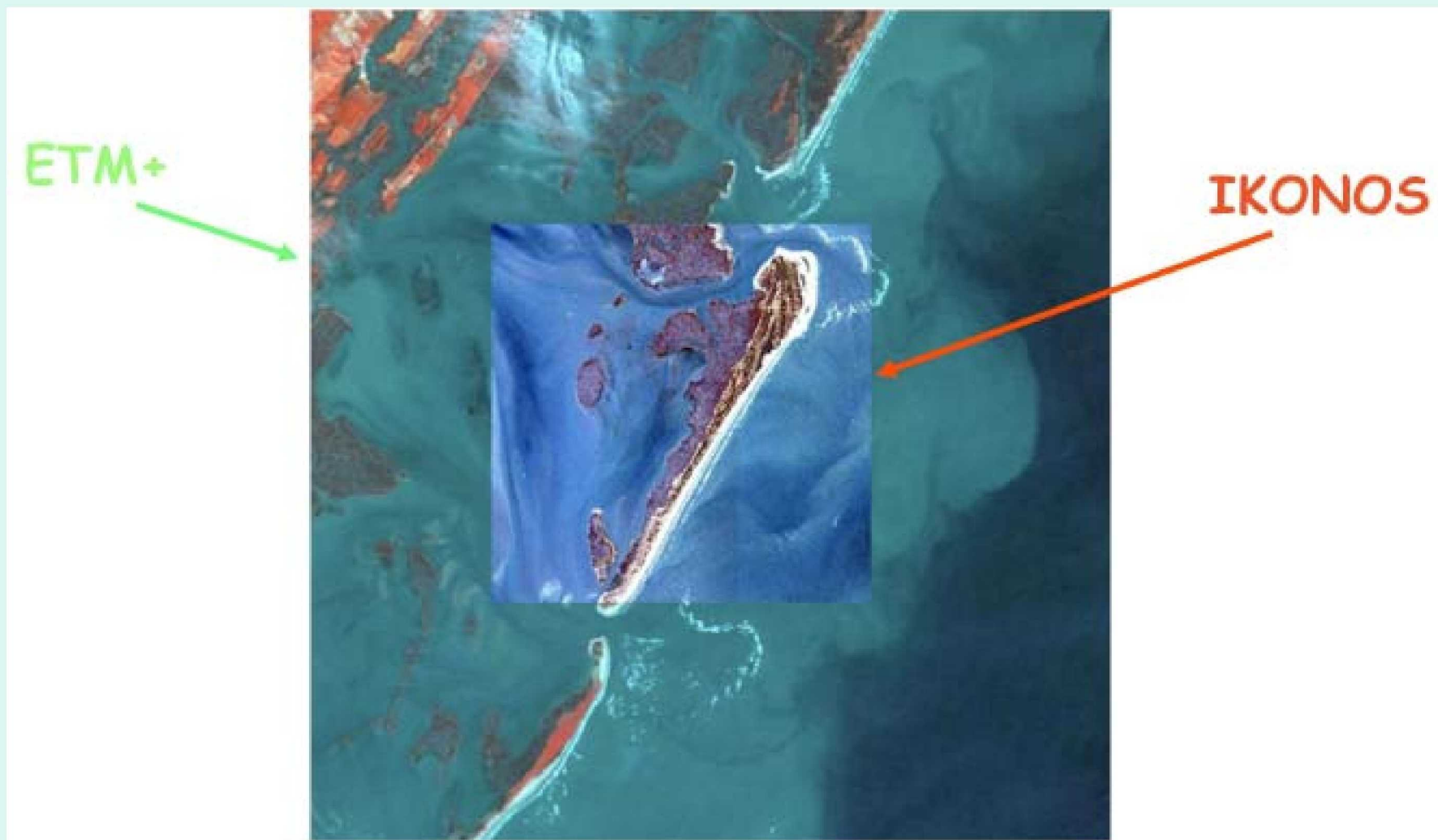
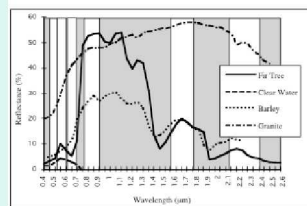
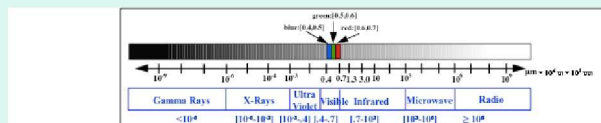
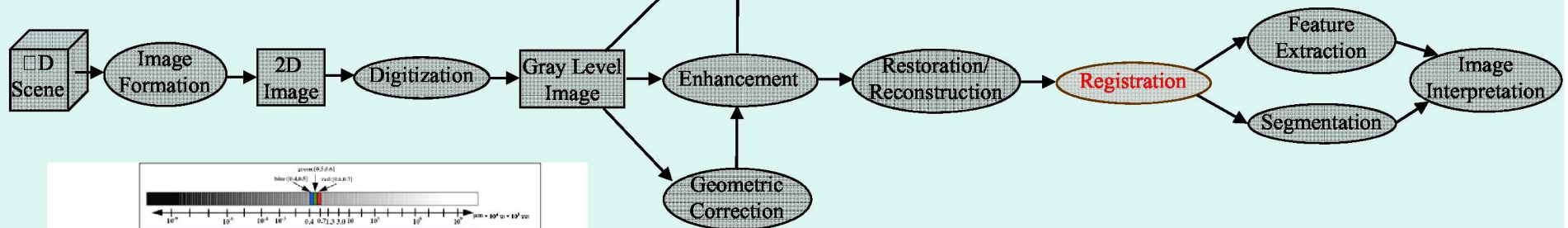
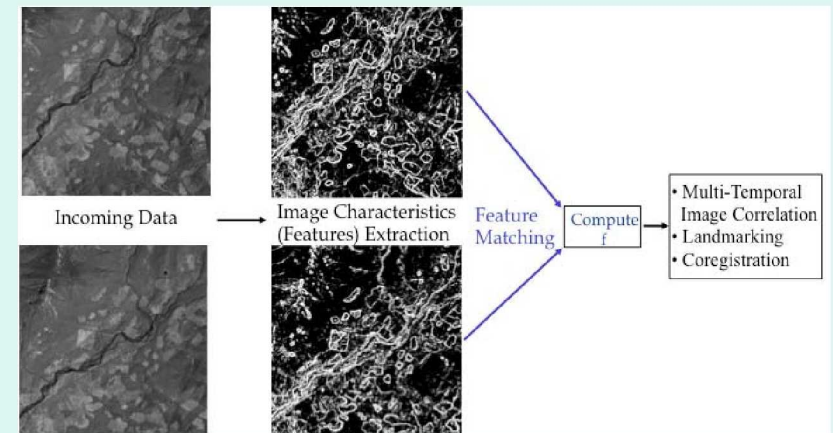
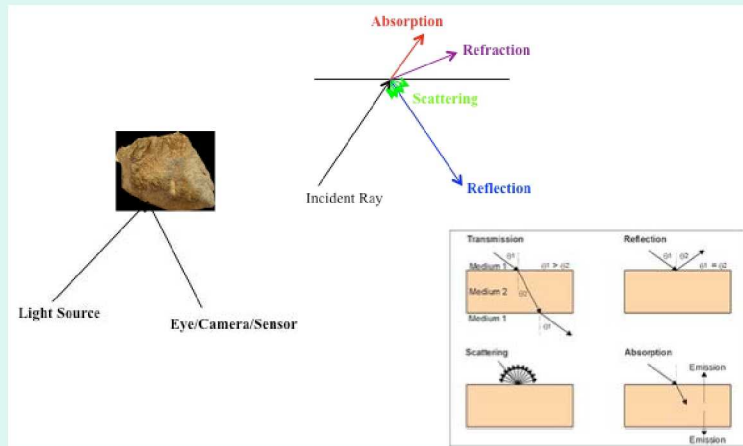


Image Processing Framework for Remotely Sensed Data



Examples of Spectral Response Patterns for 4 Different Types of Features - Fir Tree, Clear Water, Barley, Granite - White Areas Show the Portions of the Spectrum Corresponding to the 7 Channels of Landsat-Thematic Mapper (TM)

Signal to Noise at Wavelength λ :

$$(S/N)_{\lambda} = D_{\lambda} \beta^2 (H/V)^{1/2} \Delta_{\lambda} L_{\lambda}$$

Where

- D_{λ} : detectivity (measures detector performance quality)
- β : instantaneous field of view
- H : flying height of the spacecraft
- V : velocity of the spacecraft
- Δ_{λ} : spectral bandwidth of the channel (spectral resolution)
- L_{λ} : spectral radiance of ground feature

=> Tradeoff between spatial and spectral resolutions, e.g.:
To maintain the same SNR while improving spatial resolution by a factor of 8 (i.e., decreasing β by a factor of 2), we must degrade the spectral resolution by a factor of 4 (i.e., increase Δ_{λ} by a factor of 4)

The role of Image Registration in the Processing of Remotely Sensed Data

- Essential for spatial and radiometric calibration of multitemporal measurements for creating long-term phenomenon tracking data
- Used for accurate change detection:
 - (Towsn hend et al, 1992) and (Dai & Khorram, 1998): small error in registration may have a large impact on global change measurements accuracy
 - e.g., 1 pixel misregistration error => 50% error in NDVI^{*} computation (using 250m MODIS data)
- Basis for extrapolating data throughout several scales for multi-scale phenomena (distinguish between natural and human-induced)

* Normalized Difference Vegetation Index

Classifying Image Registration Utilization

- *Multimodal registration*, for integrating complementary information from multiple sensors
- *Multitemporal registration*, for change detection and Earth resource surveying
- *Viewpoint registration*, for landmark navigation, formation flying (sensor web) and planet exploration
- *Template registration*, for content-based searching or map updating

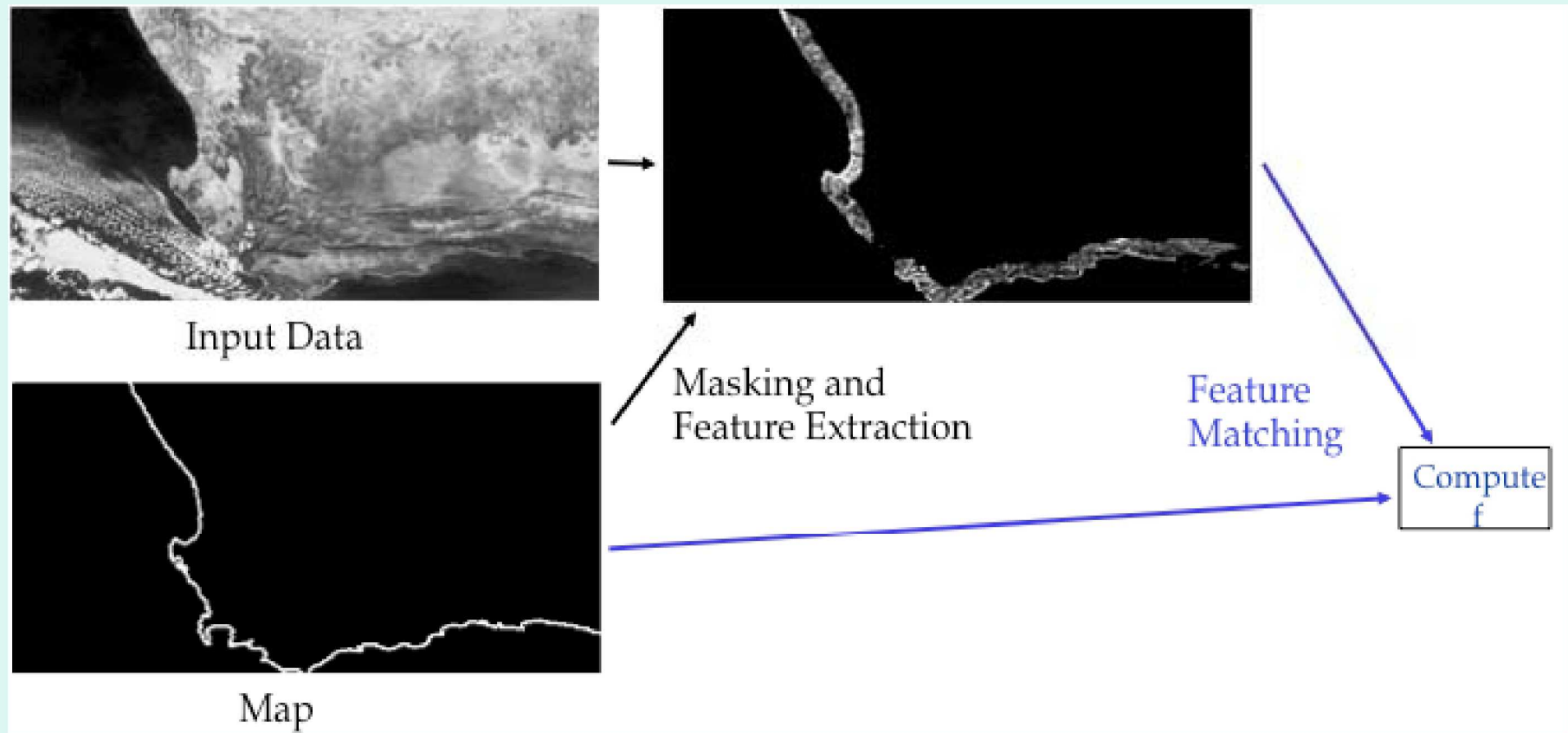
Image Registration Requirements

- *High Accuracy:* Goal of sub-pixel accuracy
- *Consistency:* Robustness to recurring use
- *Speed and High-Level of Autonomy:* Needed for
 - Large amounts of data
 - Near- or Near-real time applications (e.g., disaster management)

Systematic and Precision Corrections

- **Navigation or Model-Based Systematic Correction**
 - Orbital, attitude, platform/sensor geometric relationship, sensor characteristics, Earth model, etc.
- **Image Registration or Feature-Based Precision Correction**
 - Navigation within a few pixels accuracy
 - Image registration using selected features (or Control Points) to refine geo-location accuracy
- **Two approaches**
 1. Image registration as post-processing
 2. Navigation and image registration in a closed loop

Systematic and Precision Corrections AVHRR Example

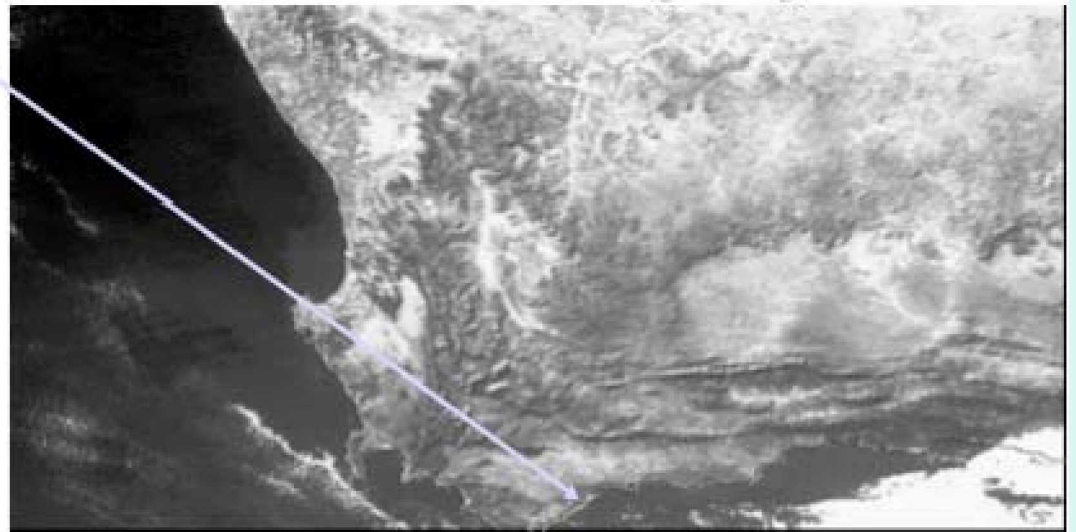


Systematic and Precision Corrections AVHRR Example (cont.)



After Navigation and
Before Image Registration

After Image Registration



Challenges in Registration of Remotely Sensed Data

- Image registration developed in other domains (medical, military, etc.) not always applicable
 - Variety in the types of sensor data and the conditions of data acquisition
 - Size of the data
 - Lack of a known image model
 - Lack of well-distributed “fiducial point” resulting in the difficulty to validate image registration methods in the remote sensing domain
 - » use synthetic data, “ground truth”, finer resolution data and “circular” registrations

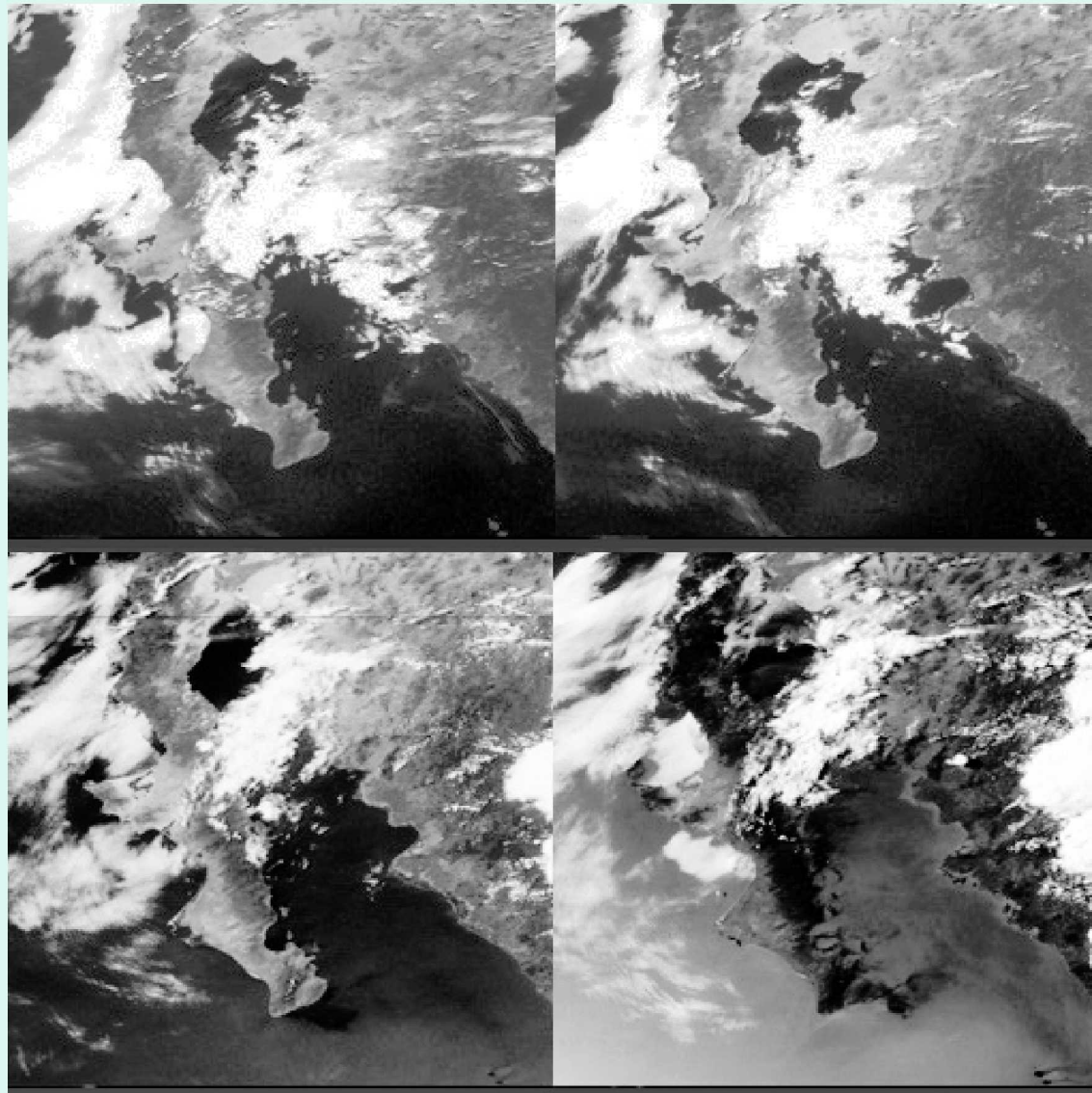
Other Challenges Facing Image Registration In the Remote Sensing Domain

- Navigation error
 - Historical satellites (e.g., Landsat-5 compared to Landsat-7)
 - Following a maneuver (e.g., star tracking)
 - Need for sub-pixel accuracy
- Atmospheric and cloud interactions
- Multitemporal effects
- Terrain/relief effect
- Multisensor data with different spatial and spectral resolutions

Atmospheric and Cloud Interactions

Baja Peninsula, California; 4 different times of the day (GOES-8)

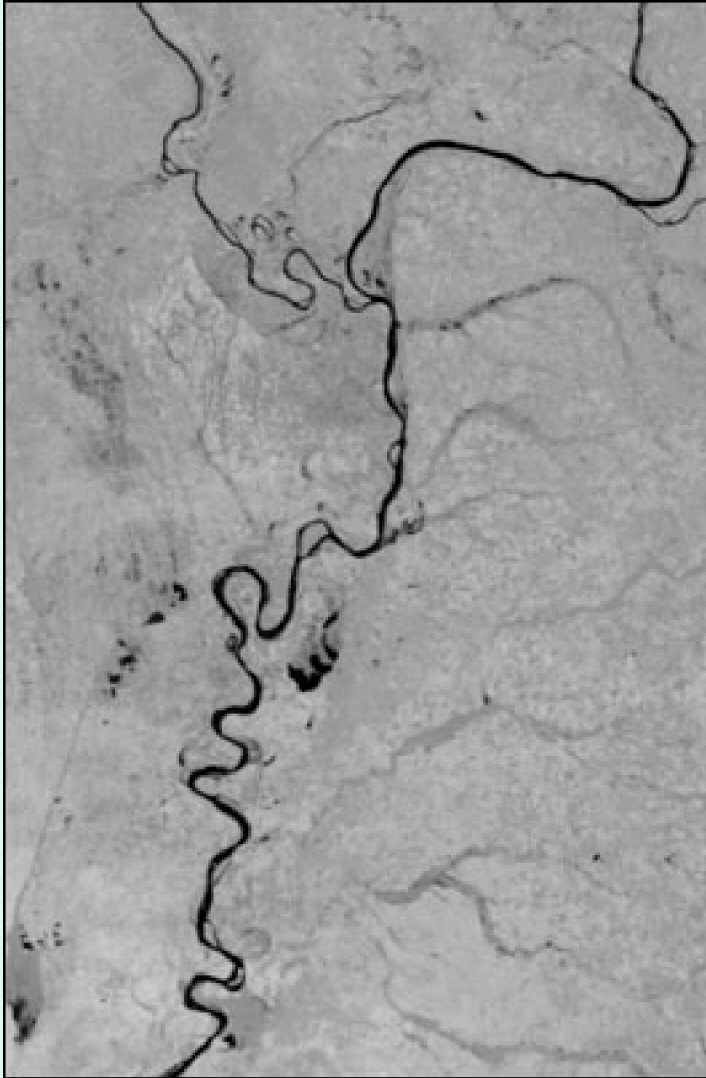
(Reproduced from Le Moigne & Eastman, 2005)



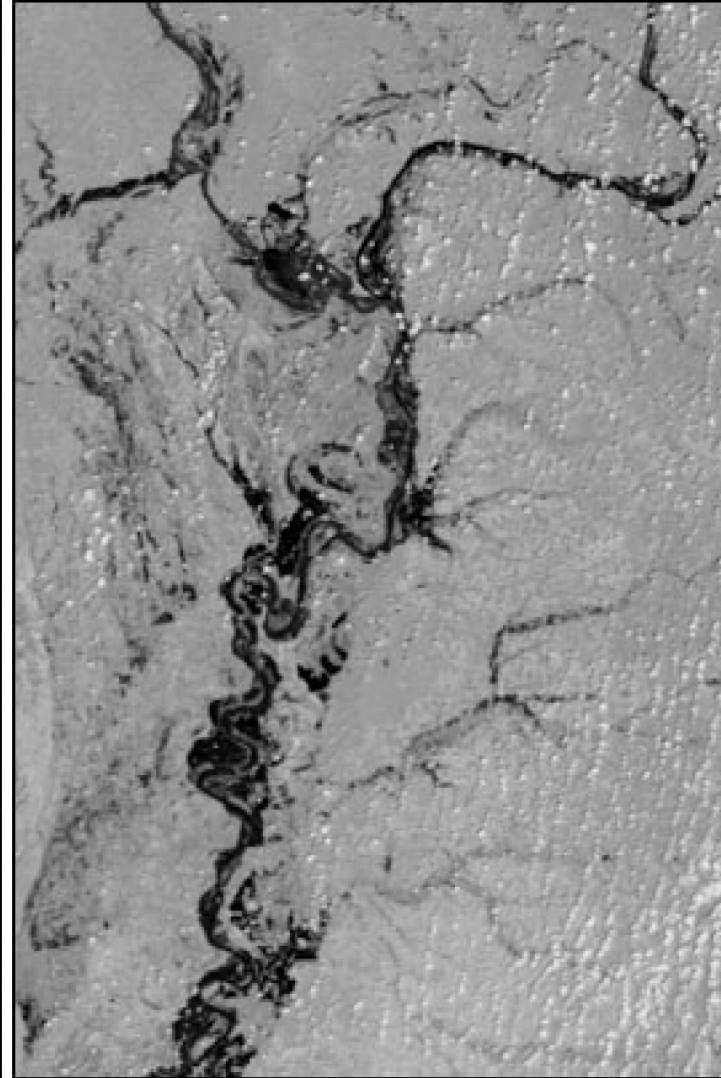
Multitemporal Effects

Mississippi and Ohio Rivers before & after Flood of Spring 2002 (Terra/MODIS)

(Reproduced from Le Moigne & Eastman, 2005)



April 25, 2002

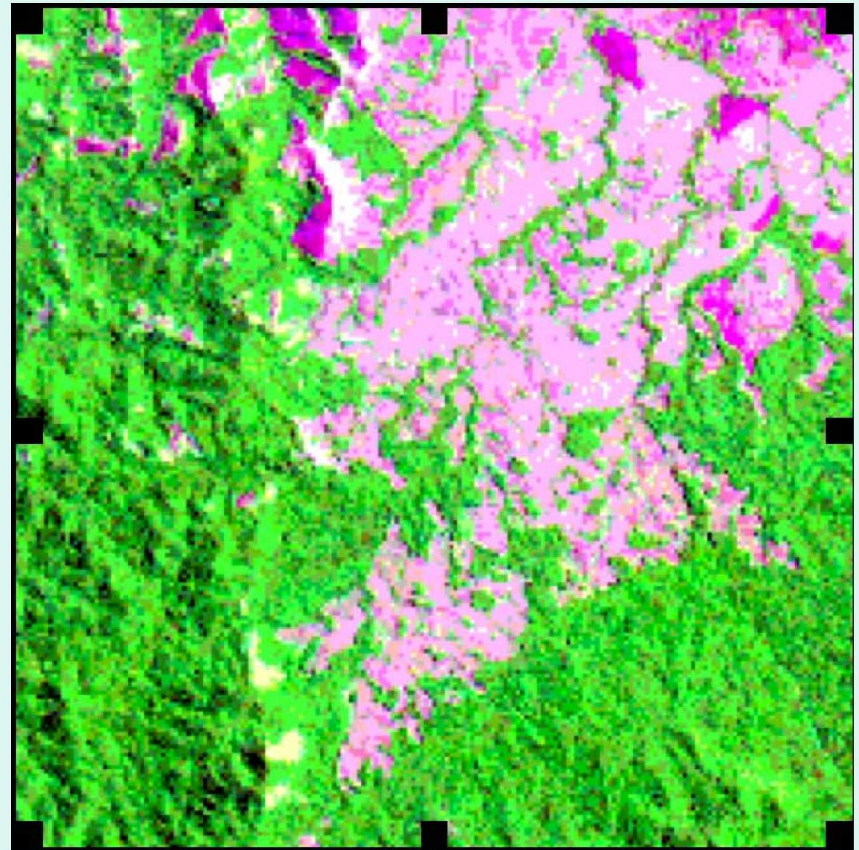
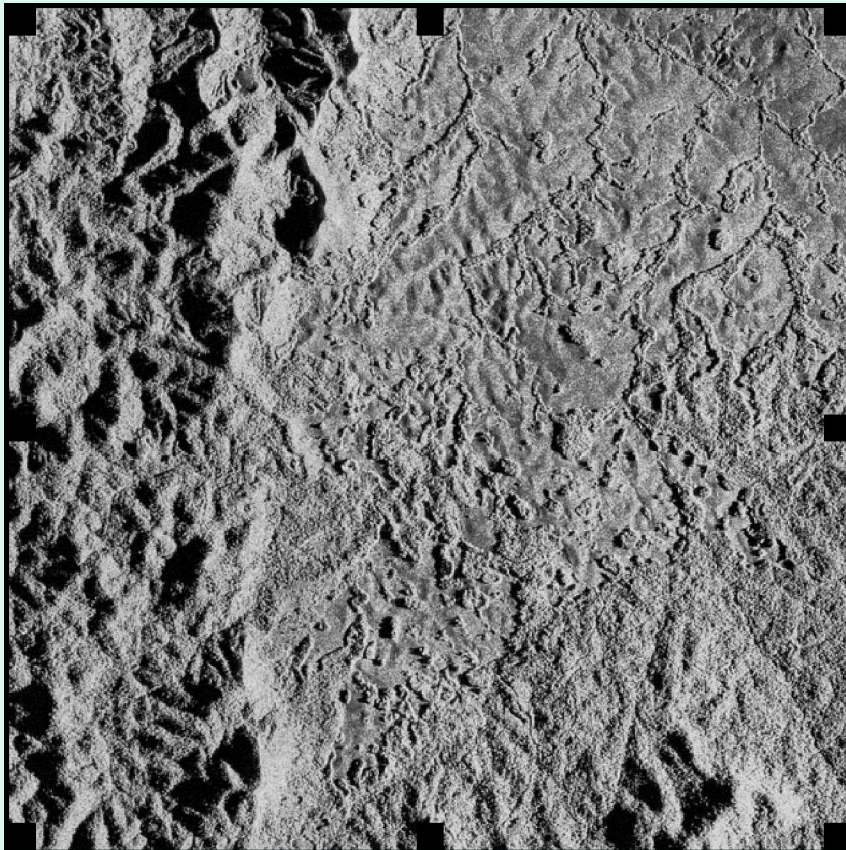


May 18, 2002

Relief Effect

SAR and Landsat-TM Data of Lopé Area, Gabon, Africa

(Reproduced from Le Moigne et al., 2001)



Precision Correction in Operational Systems

- Operational Environment

- Platform/sensor models integrated
- Historical data available for statistics/modeling
- Robustness and consistency over time is a requirement

- Operational Needs

- Systematic correction (close to 1 pixel) using navigation model
- Precision correction (less than 1 pixel) used to:
 - Check navigation model and ephemeris data
 - Perform band to band geometric calibration
 - Perform radiometric calibration of new sensor (relative to old one)

- General Characteristics

- Use database of Ground Control Points (GCP) or Chips
- Normalized Cross-Correlation (NCC) is the most common similarity measure
- Digital Elevation Model (DEM) is rarely integrated in the registration process
- Cloud masking usually integrated
- Errors in the [0.15-0.5] range

Precision Correction in Operational Systems

Some Examples - Highlights

- **AVHRR:** AUTONAV algorithm computes attitude corrections using Maximum Cross-Correlation (MCC) method between sequential images
- **GOES/METEOSAT:** CPs and NOAA Shoreline database (GSHHS) used to match edges extracted from meteorological images
- **LANDSAT:** CP image chips (1m orthorectified) using Gaussian pyramid, automatic Moravec window extraction and NCC or Mutual Information
- **MISR:** Database of 120 GCPs (each a collection of nine geolocated image patches of a well-defined and easily identifiable ground features, from Landsat, terrain-corrected, data) & ray casting simulation software
- **MODIS:** Biases and trends in the sensor orientation determined from automated control point (CP) matching and removed by updating models of the spacecraft and instrument orientation; finer CGPs from Landsat TM and ETM aggregated using PSFs and correlated with NCC
- **SEAWIFS:** Reference catalog of islands GCPs and matching using spectral classification and clustering of data, “nearest neighbor” and pattern matching techniques
- **SPOT:** Reference3DTM using DEM ortho-rectified simulated reference image in focal plane geometry, matching of input image to simulated using NCC and resampling into a cartographic reference frame
- **VEGETATION:** Database of CPs from SPOT for VEGETATION1 and VEGETATION1 for VEGETATION2; Matching by NCC

Image Registration at NASA GSFC

- Operational Environment

- Platform/sensor models integrated
- Historical data available for statistics/modeling
- Robustness and consistency over time is a requirement

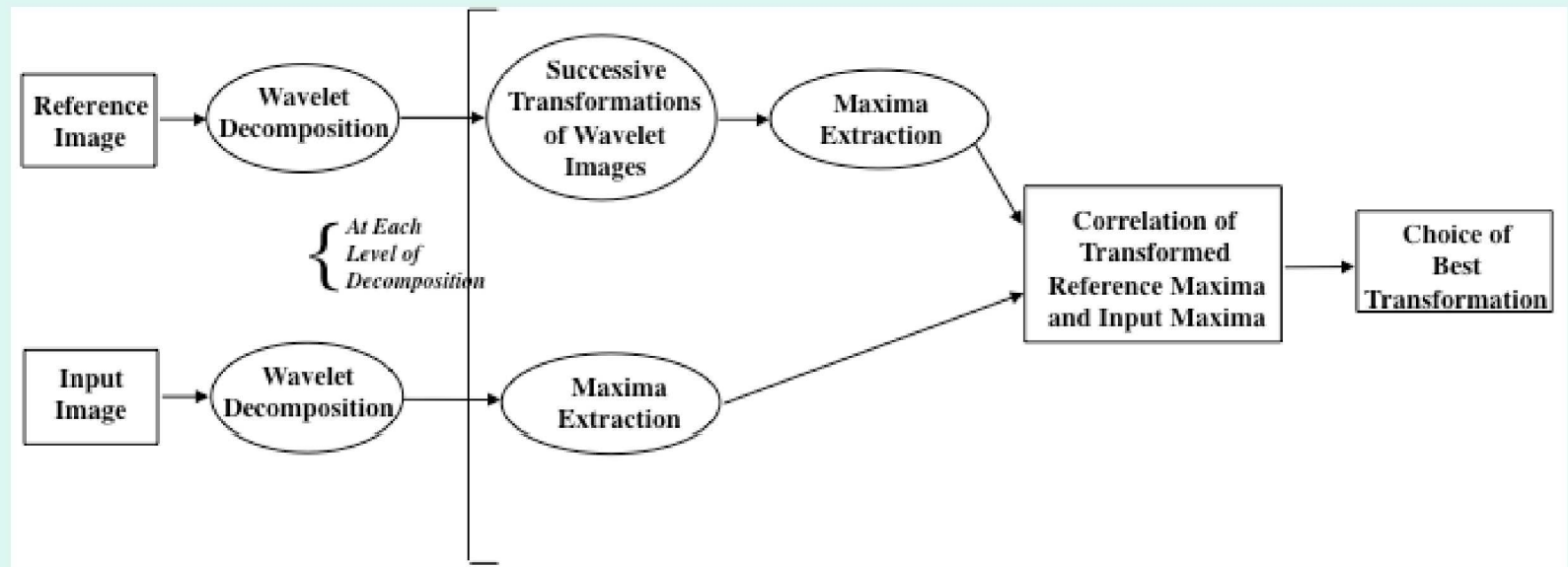
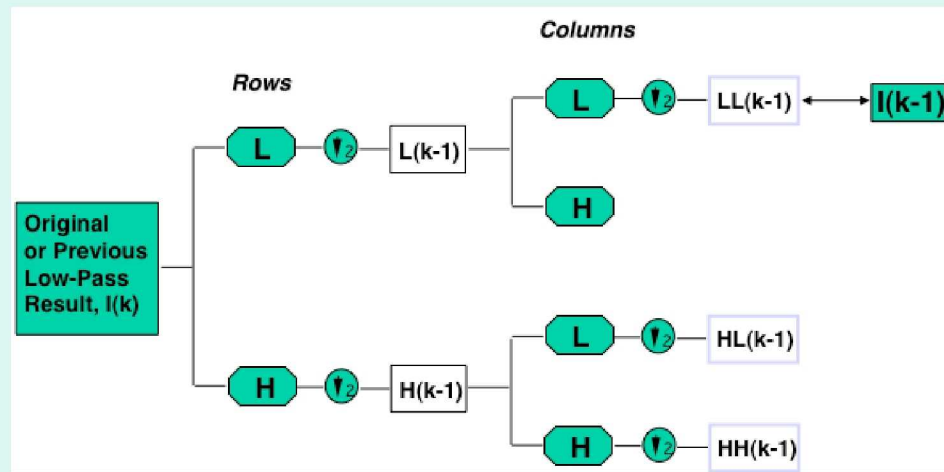
- Operational Needs

- Systematic correction (close to 1 pixel) using navigation model
- Precision correction (less than 1 pixel) used to:
 - Check navigation model and ephemeris data
 - Perform band to band geometric calibration
 - Perform radiometric calibration of new sensor (relative to old one)

- General Characteristics

- Use database of Ground Control Points (GCP) or Chips
- Normalized Cross-Correlation (NCC) is the most common similarity measure
- Digital Elevation Model (DEM) is rarely integrated in the registration process
- Cloud masking usually integrated
- Errors in the [0.15-0.5] range

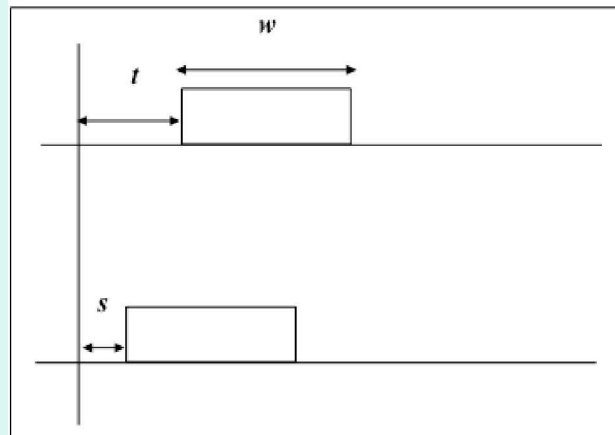
Orthogonal Wavelet Image Registration



Orthogonal Wavelet Image Registration

Rotation and Translation Invariance Issues

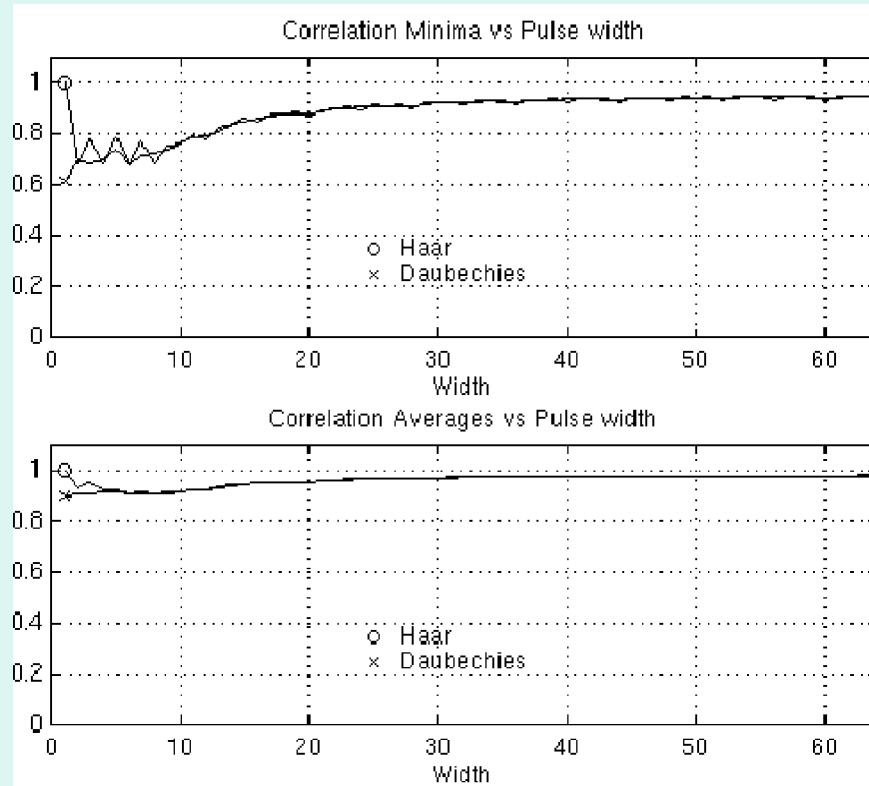
- Nyquist criterion, sample signal at least twice frequency of highest frequency component
 - in og wavelets, signal changes within or across subbands with subsampling
- Study for Shift Sensitivity (Stone et al, 1999):
 - low-pass subband relatively insensitive to translation, if features are twice the size of wavelet filters
 - high-pass subband more sensitive but can still be used.



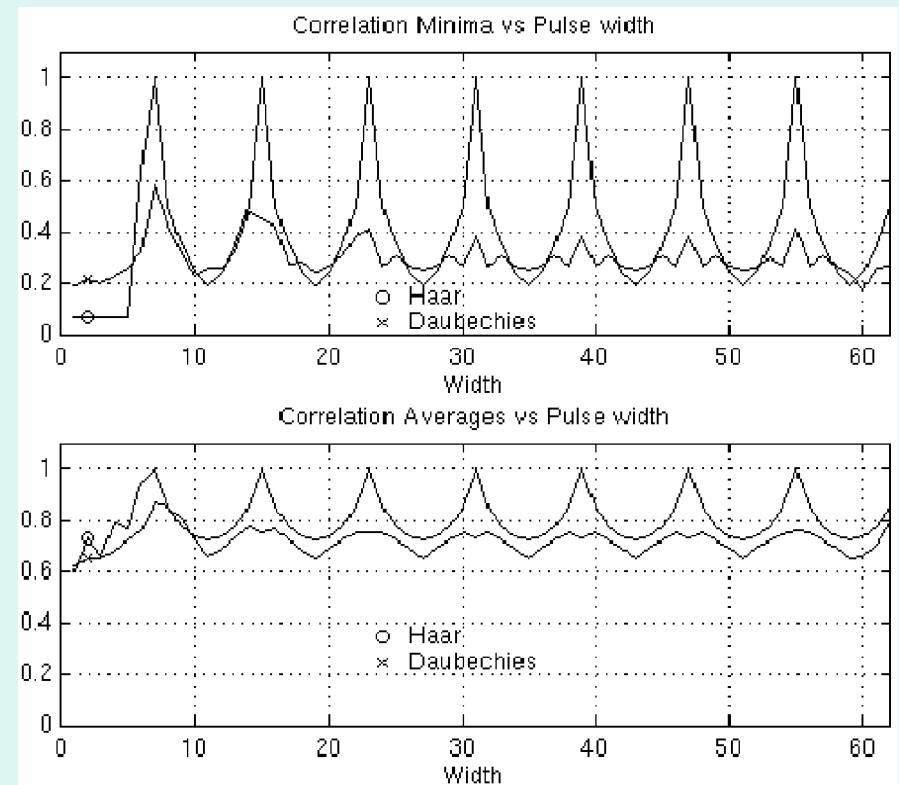
Correlate Wavelets of Two Pulses

Orthogonal Wavelet Image Registration

Rotation and Translation Invariance Issues (cont.)



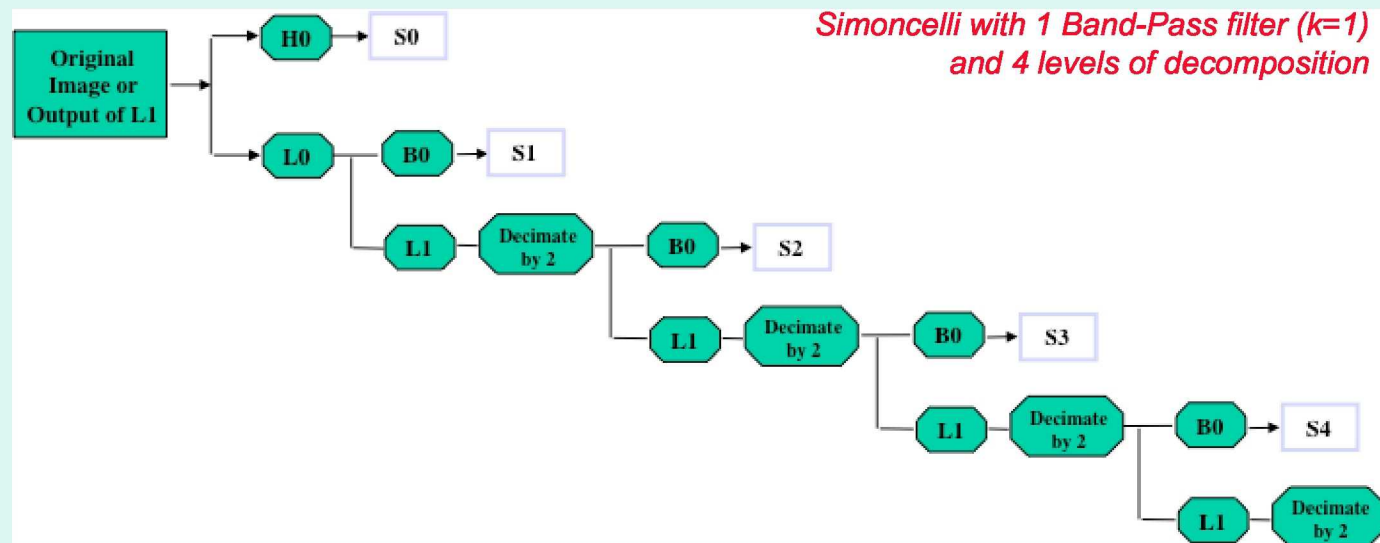
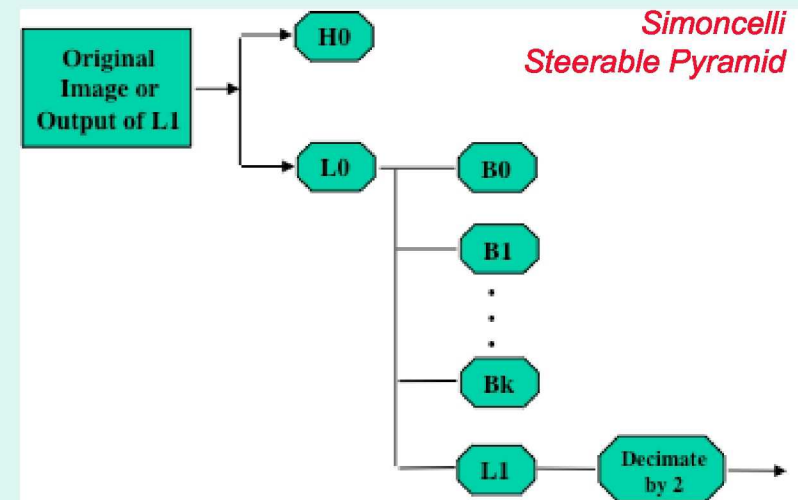
Translation Sensitivity Low-Pass Level 3



Translation Sensitivity High-Pass Level 3

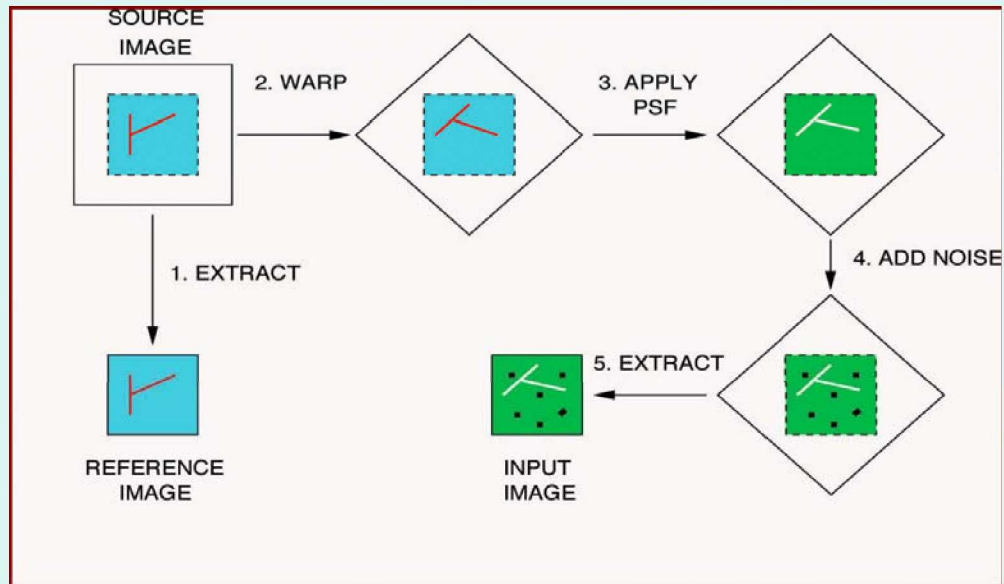
Rotation- and Translation-Invariant Pyramids

- Simoncelli:
 - Relax critical sampling condition of wavelet transforms
 - Overcomplete representation by $4k/3$ (k: number of band-pass filters)
- Splines:
 - Recursive anti-aliasing prefiltering followed by a decimation of 2
 - Only low-pass bands

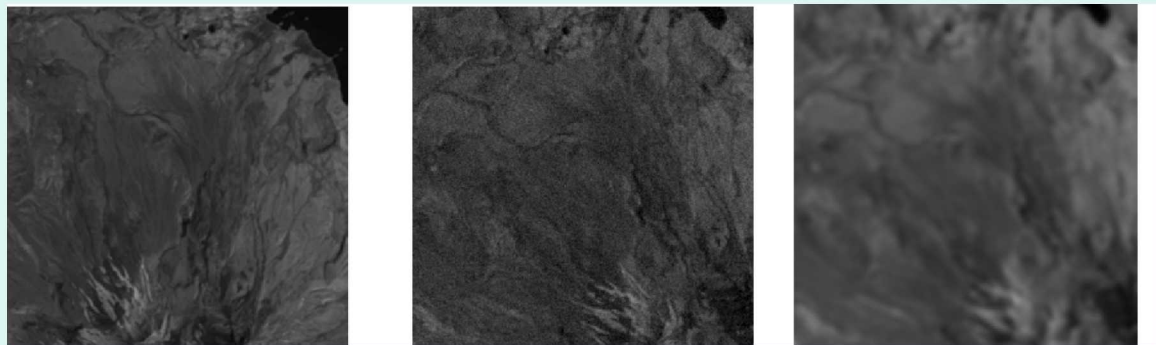


Comparative Studies Using Synthetic Data

(Reproduced from Zavorin & Le Moigne, 2005)



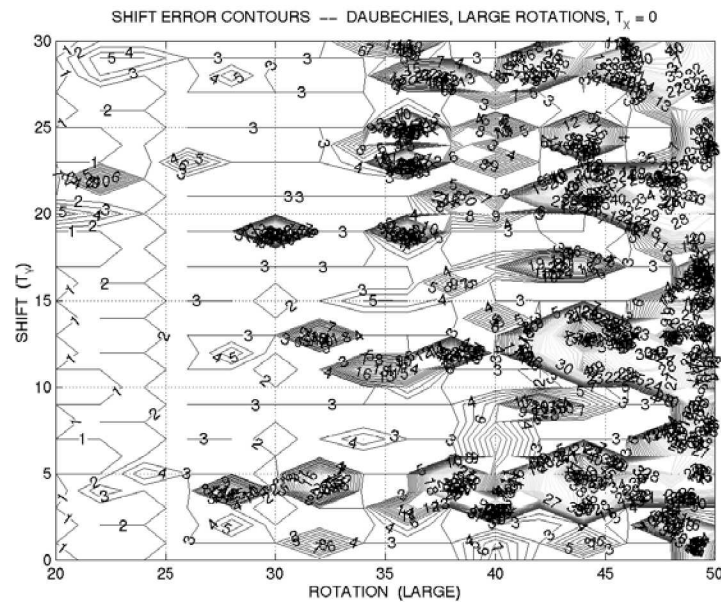
Synthetic Image Generation



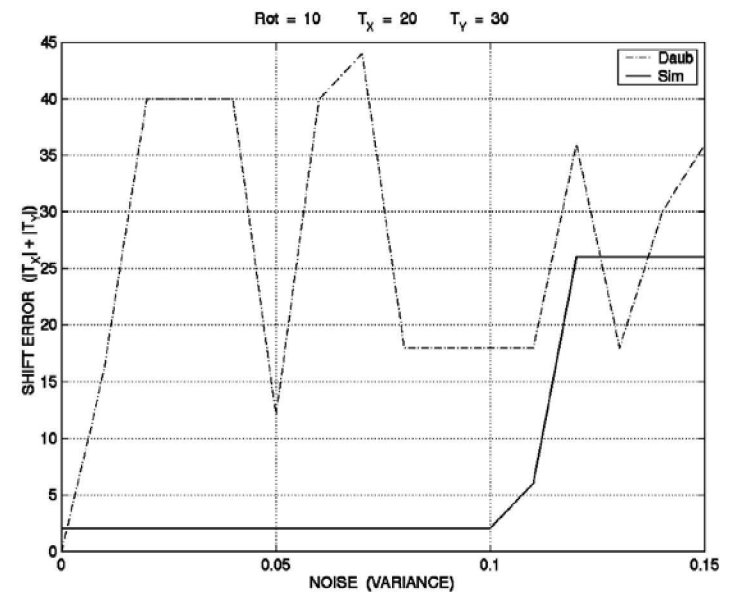
Synthetic Image Examples (Original; Warp & Noise; Warp & PSF)

Orthogonal Wavelet Studies

(Reproduced from Le Moigne & Zavorin, 2000)



Shift Errors – Daubechies – Large Rotations



*Shift Errors Function of Noise – Daubechies
Large Rotation and Translation*

Spline and Simoncelli Pyramids Studies

(Reproduced from Zavorin & Le Moigne, 2005)

	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	1657/7236 \approx 22.9%	0.0219008	0.086284	0.148911
TRU-SimB	1552/7236 \approx 21.5%	0.0411122	0.141976	0.212933
TRU-SimL	3693/7236 \approx 51%	0.0214522	0.056263	0.109165

Average Error for Converged Region of Test Dataset (Warp & Noise)

	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	2831/9801 \approx 28.9%	0.285565	0.300596	0.082751
TRU-SimB	725/9801 \approx 7.4%	0.052036	0.065967	0.032609
TRU-SimL	2918/9801 \approx 29.8%	0.320529	0.331067	0.091969

Average Error for Converged Region of Test Dataset (Warp & PSF)

	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	1424/7236 \approx 19.7%	0.216392	0.278916	0.168494
TRU-SimB	1415/7236 \approx 19.6%	0.164233	0.252788	0.213441
TRU-SimL	4038/7236 \approx 55.8%	0.243106	0.289142	0.142683

Average Error for Converged Region of Test Dataset (Warp & PSF & Noise)

Spline and Simoncelli Pyramids Studies

(Reproduced from Zavorin & Le Moigne, 2005)

	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	1657/7236 \approx 22.9%	0.0219008	0.086284	0.148911
TRU-SimB	1552/7236 \approx 21.5%	0.0411122	0.141976	0.212933
TRU-SimL	3693/7236 \approx 51%	0.0214522	0.056263	0.109165

Average Error for Converged Region of Test Dataset (Warp & Noise)

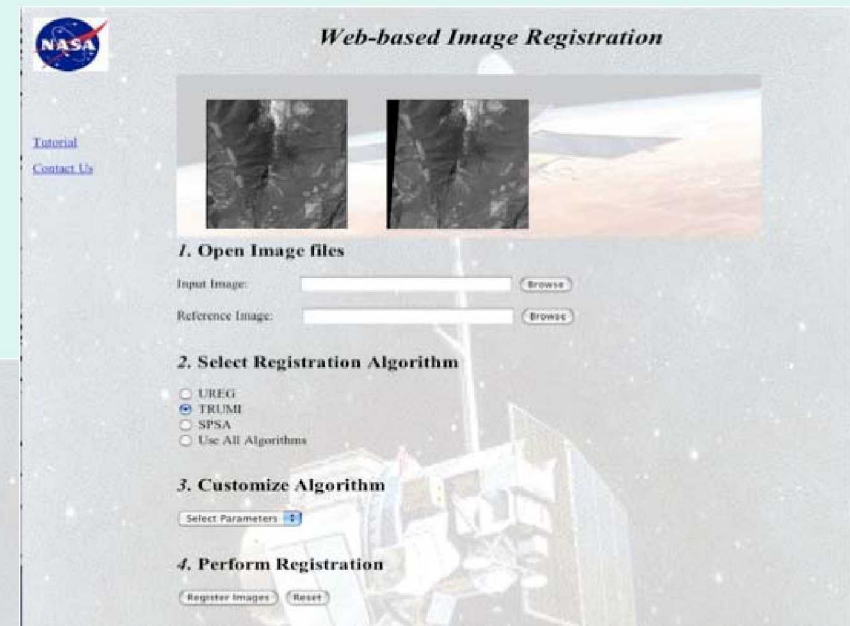
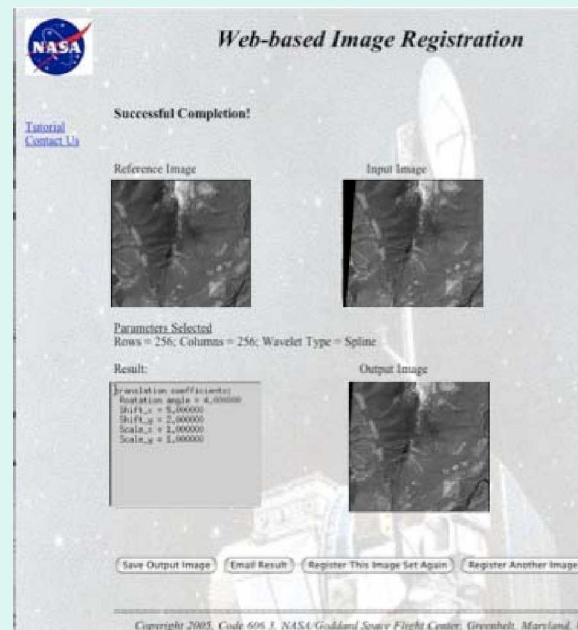
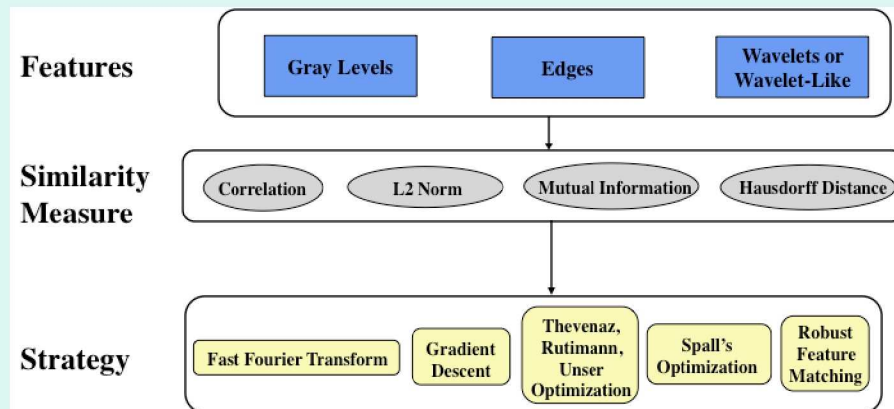
	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	2831/9801 \approx 28.9%	0.285565	0.300596	0.082751
TRU-SimB	725/9801 \approx 7.4%	0.052036	0.065967	0.032609
TRU-SimL	2918/9801 \approx 29.8%	0.320529	0.331067	0.091969

Average Error for Converged Region of Test Dataset (Warp & PSF)

	<i>Number of converged</i>	<i>Median converged error</i>	<i>Mean converged error</i>	<i>Standard deviation converged error</i>
TRU-SplC	1424/7236 \approx 19.7%	0.216392	0.278916	0.168494
TRU-SimB	1415/7236 \approx 19.6%	0.164233	0.252788	0.213441
TRU-SimL	4038/7236 \approx 55.8%	0.243106	0.289142	0.142683

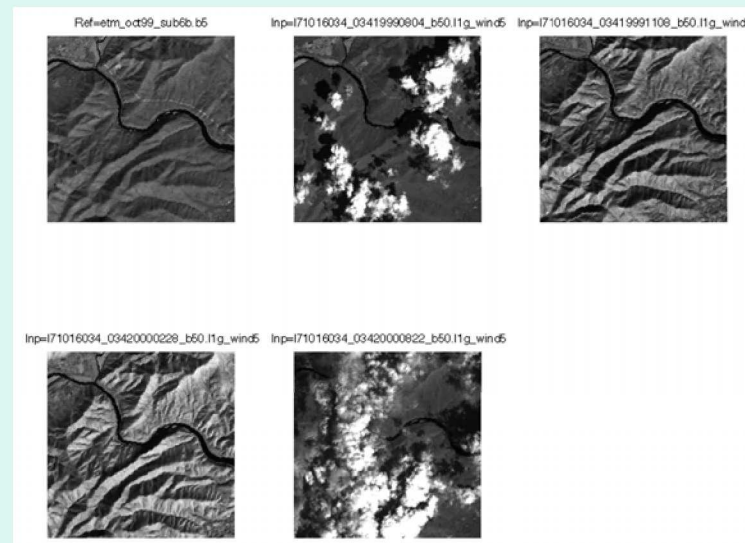
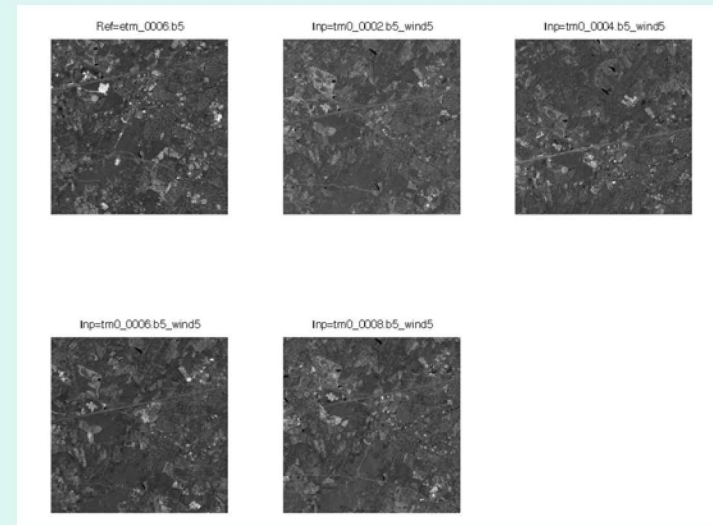
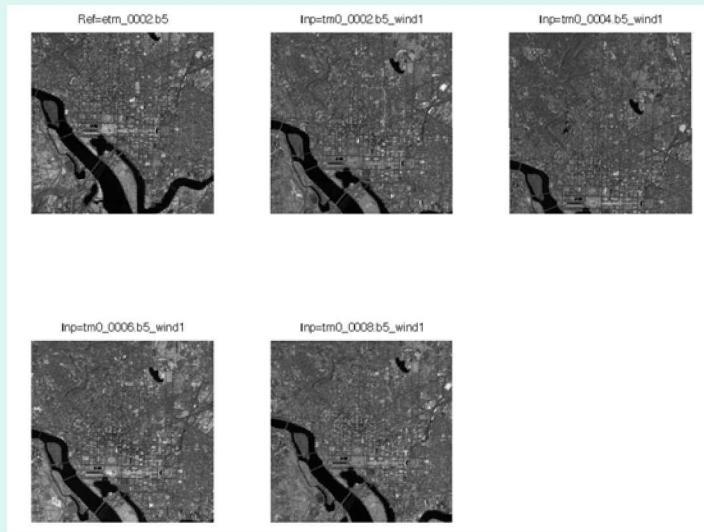
Average Error for Converged Region of Test Dataset (Warp & PSF & Noise)

A Framework for the Analysis of Various Image Registration Components



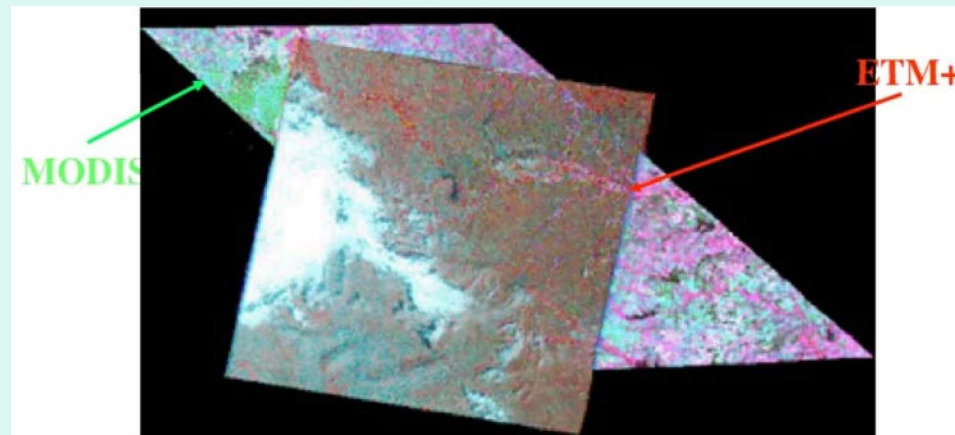
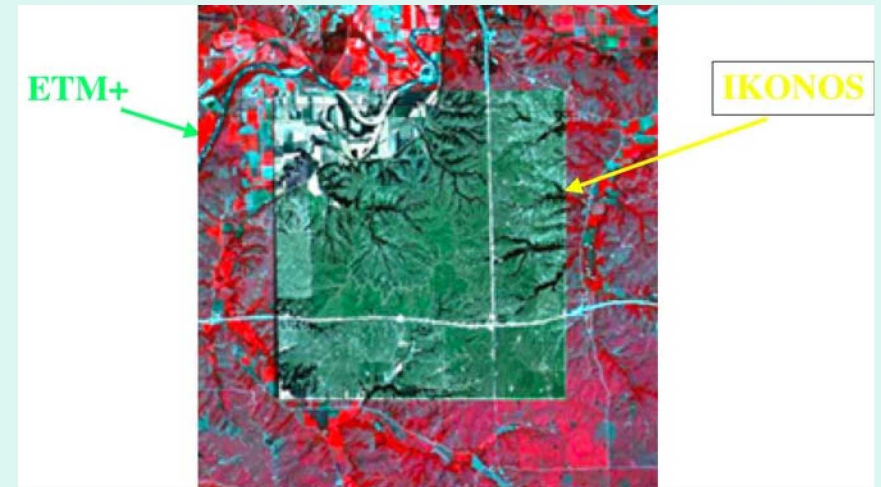
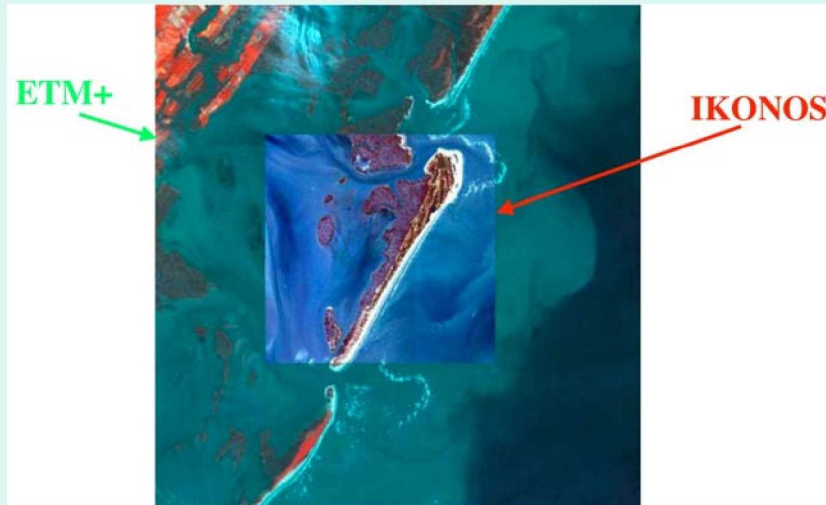
TARA (Toolbox for Automated Registration and Analysis)

Algorithm Testing Using Landsat-TM Multitemporal Data



Algorithm Testing Using Multisensor Data (ETM, IKONOS and MODIS)

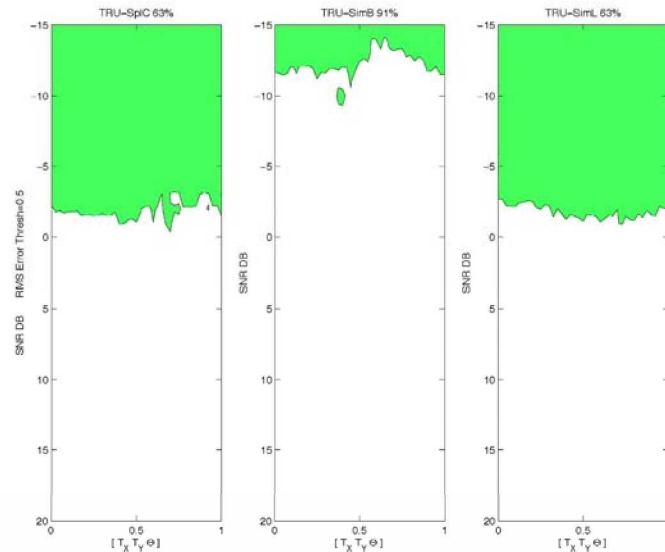
Red and NIR Bands; 30m – 4m – 250 and 500 m respectively



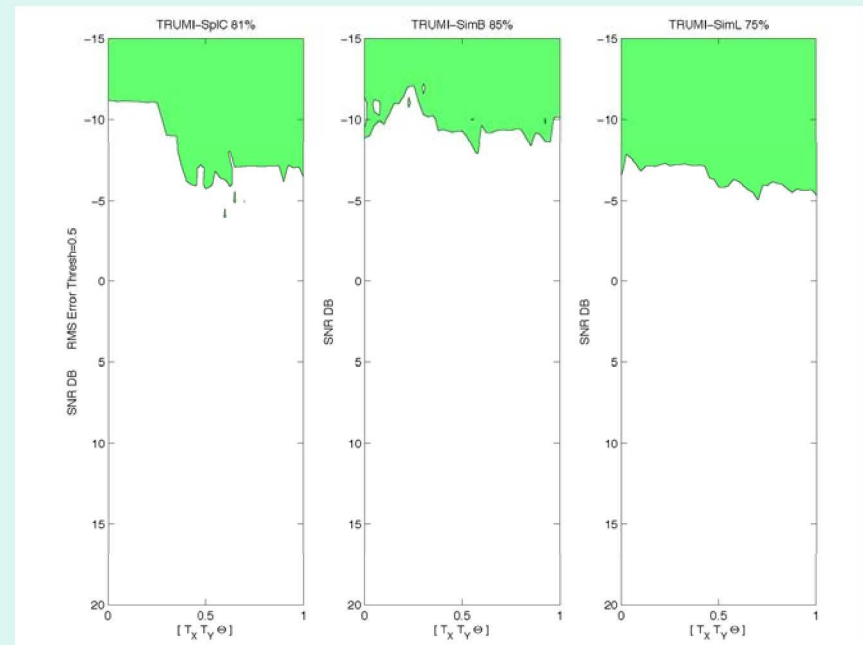
Framework Testing Using Synthetic Datasets

Marquart-Levenberg Optimization Using L2-Norm and Mutual Information

(Reproduced from Zavorin & Le Moigne, 2005)



*Contour Plot "SameRadNoisy" Dataset
Optimization Using L2-Norm
Threshold of 0.5*

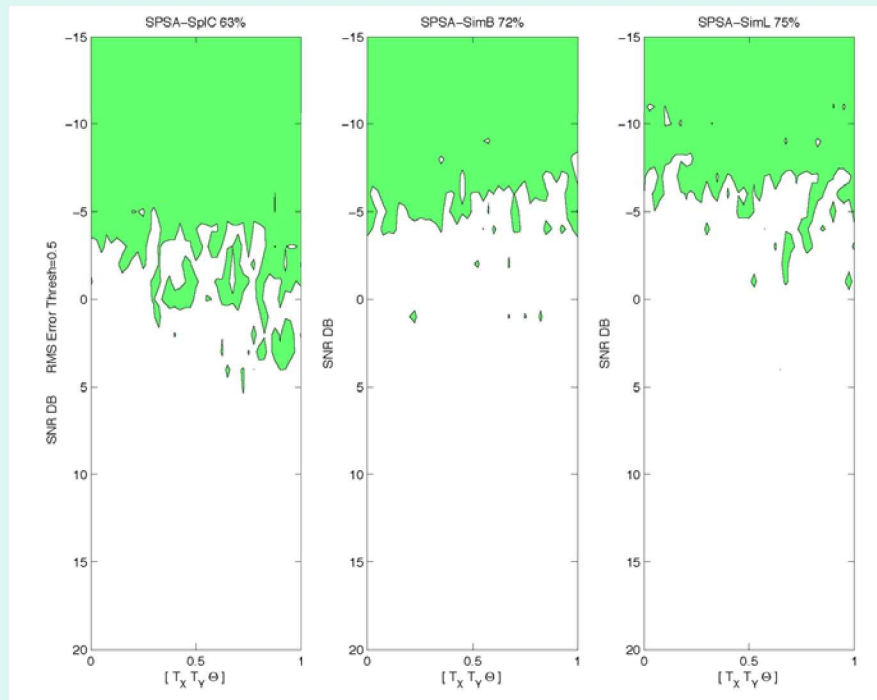


*Contour Plot "SameRadNoisy" Dataset
Optimization Using Mutual Information
Threshold of 0.5*

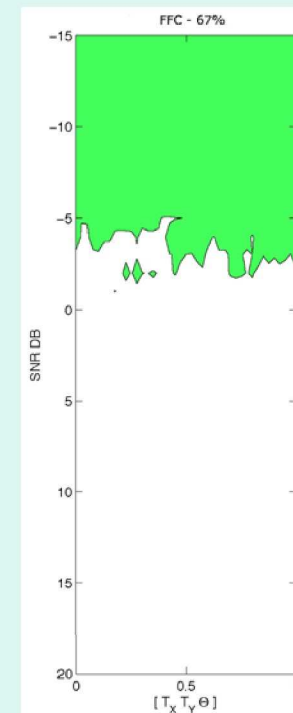
Framework Testing Using Synthetic Datasets

Stochastic Gradient Optimization Using L2-Norm and Mutual Information

(Reproduced from Zavorin & Le Moigne, 2005)



*Contour Plot “SameRadNoisy” Dataset
Stochastic Gradient and Mutual Information
Threshold of 0.5*



*Contour Plot “SameRadNoisy” Dataset
Fast Fourier Correlation
Threshold of 0.5*

Multitemporal Datasets

Robust Feature Matching Using Simoncelli Band-Pass Features

(Reproduced from Netanyahu et al, 2004)

	RFM REGISTRATION			MANUAL GROUND TRUTH			ABSOLUTE ERROR		
Scene	Q	T _x	T _y	Q	T _x	T _y	DQ	DT _x	DT _y
840827	0.031	4.72	-46.88	0.026	5.15	-46.26	0.005	0.43	0.62
870516	0.051	8.49	-45.62	0.034	8.58	-45.99	0.017	0.09	0.37
900812	0.019	17.97	-33.36	0.029	15.86	-33.51	0.010	0.11	0.15
960711	0.049	8.34	- 101.97	0.031	8.11	- 103.18	0.018	0.23	1.21

*Results of Multitemporal Registration
Using Landsat-TM Data over DC/Baltimore Area*

	RFM REGISTRATION			MANUAL GROUND TRUTH			ABSOLUTE ERROR		
Scene	Q	T _x	T _y	Q	T _x	T _y	DQ	DT _x	DT _y
990804	0.009	0.36	3.13	0.002	0.04	3.86	0.011	0.40	0.73
991108	0.000	1.00	13.00	0.002	1.20	13.53	0.002	0.20	0.53
000228	0.005	0.88	-2.32	0.008	1.26	2.44	0.003	0.38	0.12
000822	0.002	0.41	9.22	0.011	0.35	9.78	0.013	0.06	0.56

*Results of Multitemporal Registration
Using Landsat-TM Data over Virginia Area*

Multisensor Datasets

All Algorithm Comparison

(Reproduced from Le Moigne et al, 2001)

PAIR TO REGISTER	FFC		GGD		SIMB-CORREL		SIMB-MI		RFM	
	Rot	Transl	Rot	Transl	Rot	Transl	Rot	Transl	Rot	Transl
ETM_nir/ETM_red	Rotation = 0, Translation = (0,0) computed by all methods, using 7 sub-window pairs									
IKO_nir/ETM_nir	-	(2, 1)	0.0001	(1.99,-0.06)	0	(2, 0)	0	(2, 0)	0.00	(0.0, 0.0)
IKO_red/ETM_red	-	(2, 1)	-0.0015	(1.72, 0.28)	0	(2, 0)	0	(2, 0)	0.00	(0.0, 0.0)
ETM_nir/MODIS_nir	-	(-2,-4)	0.0033	(-1.78,-3.92)	0	(-2, -4)	0	(-2, -4)	0.00	(-3.0,3.5)
ETM_red/MODIS_red	-	(-2,-4)	0.0016	(-1.97,-3.90)	0	(-2, -4)	0	(-2, -4)	0.00	(-2.0,-3.5)
MODIS_nir/SEAWIFS	-	(-9, 0)	0.0032	(-8.17, 0.27)	0	(-8, 0)	0	(-9, 0)	0.50	(-6.0,2.0)
MODIS_red/SEAWIFS	-	(-9, 0)	0.0104	(-7.61, 0.57)	0	(-8, 0)	0	(-8, 0)	0.25	(-7.0,1.0)

*Results of Multisensor Registration
Using ETM, IKONOS and MODIS Data over Konza Agricultural Area*

- Similar Tests performed on:
 - Urban Area (USDA site; Greenbelt, MD)
 - Coastal Area (VA Coast)
 - Agricultural Area (Cascades Site, CO)
 - Mountainous Area (Konza Prairie, Kansas)
- Consistency studies show between 0.125 and 0.25 pixel errors using circular registrations of IKONOS NIR and Red data
- Additional studies performed on EO1-Hyperion data

Fusion of Remotely Sensed Data

- Data Fusion

- Use multi-source data of **different natures** to increase quality of information contained in data (Pohl and Genderen, 1998)
- A **process** dealing with **association, correlation, and combination** of data and information from single and **multiple** sources to achieve **refined position and identity estimates**, and complete and timely assessments of situations and threats, and their significance (Hall and Llinas, 2001).

- Image Fusion

- Data are images
- General Objectives:
 - » Image sharpening
 - » Improving registration/classification accuracy
 - » Temporal change detection
 - » Feature enhancement
- Example Application
 - » Invasive Species Forecasting System
 - » Objective
 - » **Improvement of classification accuracy**
 - » **Tamarisk, Leafy Spurge, Cheat grass, Russian olive, etc.**
 - » **Feature enhancement**

Image Fusion Methods

- Principal Component Analysis, PCA
 - Input
 - Multivariate data set of inter-correlated variables
 - Output
 - Data set of new uncorrelated linear combinations of the original variable
- Wavelet-based Fusion
 - Use of Different Subbands in Reconstruction
- Cokriging

Image Fusion Methods

Wavelet-Based Image Fusion

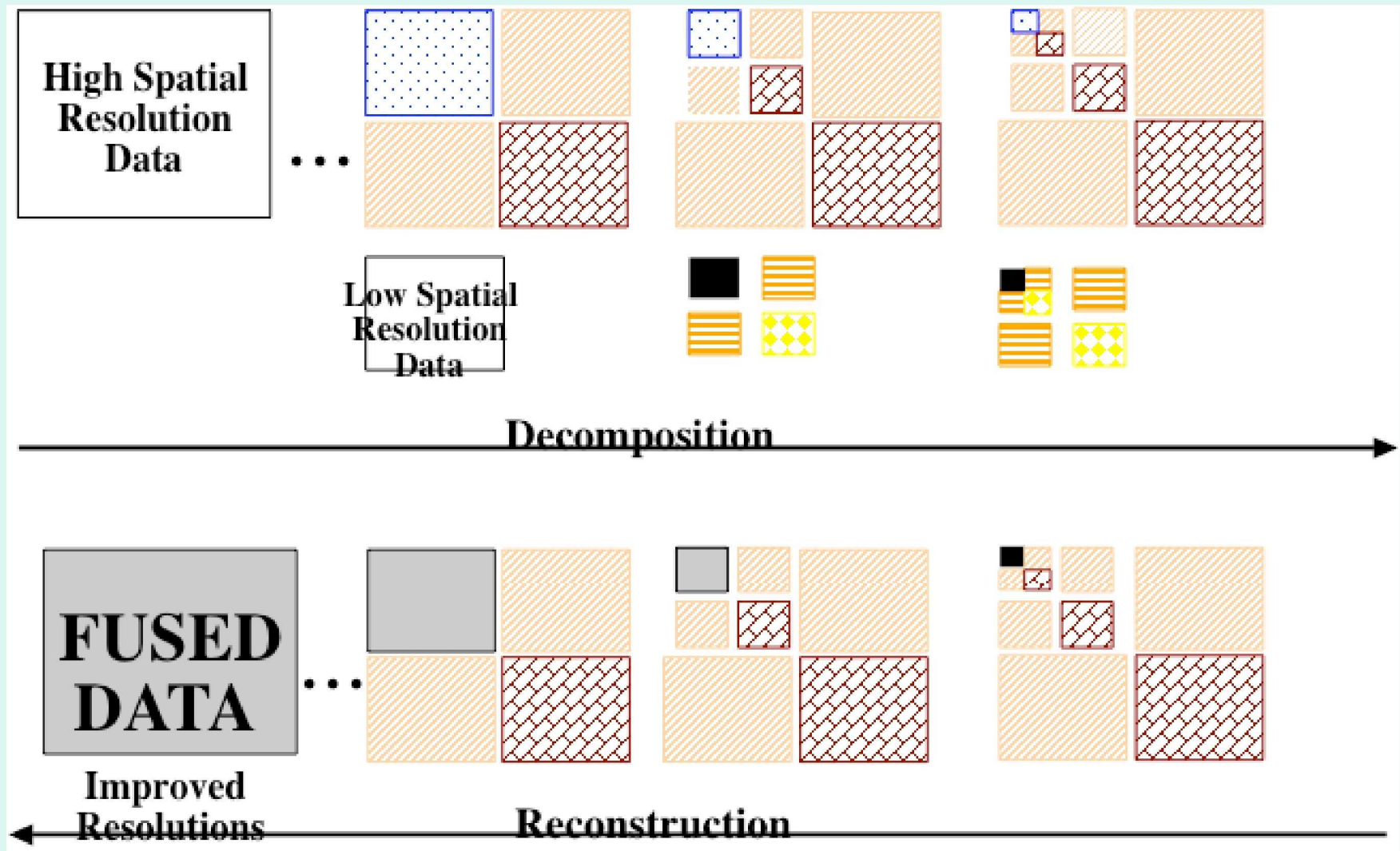


Image Fusion Methods

Cokriging

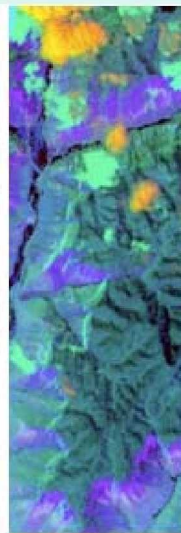
- Interpolation Method
 - Geo-statistics, mining, and petroleum engineering applications (pioneered by Danie Krige, 1951)
 - Generalized version of *kriging* (*B.L.U.E*):
 - *Best*: aims to minimize variance of the errors
 - *Linear*: estimates are weighted linear combination of the available data
 - *Unbiased*: tries to have mean residual, or error, equal to zero.
 - *Estimator*
- Interpolation using more than one type of variable to estimate an unknown value at a particular location
- Goal of cokriging is to *minimize variance of error* subject to some constraints (to ensure unbiasedness of our estimate)

Image Fusion Experiments

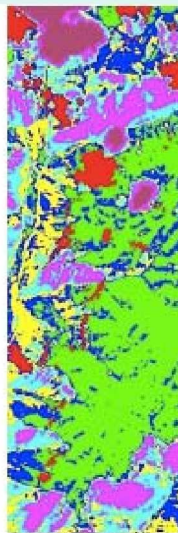
Using Principal Component Analysis

(Reproduced from Memarsadeghi et al, 2005)

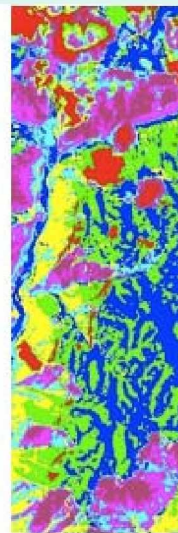
- Input
 - 9 bands of ALI
 - 140 bands of Hyperion (calibrated and not corrupted bands)
 - Stack of both ALI and Hyperion bands above
- Output
 - Same number of PCs as input bands
 - Select PCs containing 99% of information



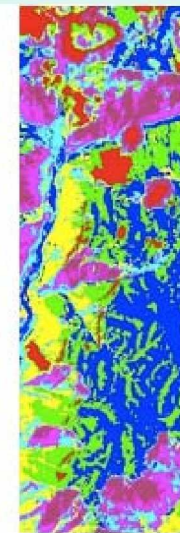
ALI PCs 1,2,3



ALI PCs 1,2,3
clustering



Hyperion
First 7 PCs
clustering



ALI-Hyp First 9 PCs
clustering

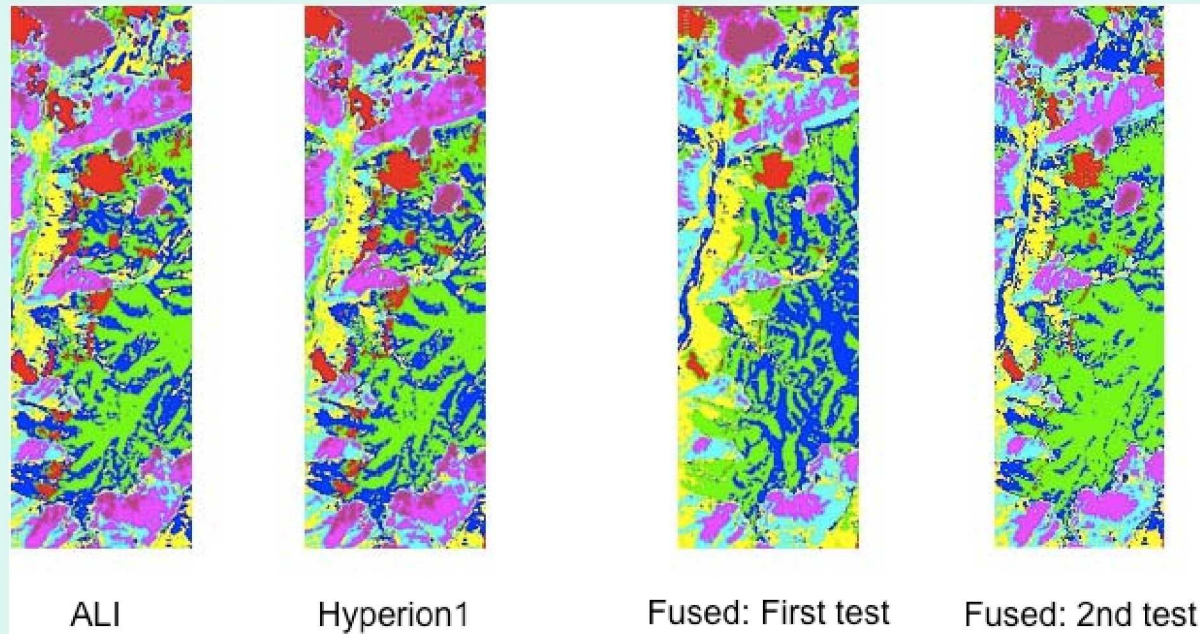
ALI V	Hyp V	Fused V
143.98	137.64	180.10

Image Fusion Experiments

Using Wavelet-Based Fusion

(Reproduced from Memarsadeghi et al, 2005)

- Fuse each multispectral band of ALI with one band of Hyperion
 - For each of 9 ALI bands
 - Select a Hyperion band within the wavelength range of corresponding ALI band which is
 - » closest to the center of ALI's wavelength range (experiment 1)
 - » least correlated to the corresponding ALI band (experiment 2)
 - Clustering of fusion result of 9 bands of ALI with 9 bands of Hyperion
 - Fusion: 4 Levels of Decomposition, Daubechies Filter of size 2

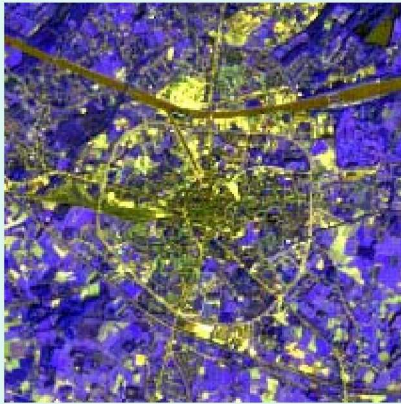


- Experiment 1 Variances: ALI: 179.73; Hyperion: 159.96; Fused: 195.27
- Experiment 2 Variances: ALI: 179.73; Hyperion: 165.34; Fused: 173.77

Image Fusion Experiments

Using Cokriging

(Reproduced from Memarsadeghi et al, 2006)



**Landsat-TM
Multispectral Bands 2, 3, 4
(30m resolution)**



**Landsat-TM Panchromatic
(15m resolution)**

Image Fusion Experiments

Using Cokriging (cont.)

(Reproduced from Memarsadeghi et al, 2006)

*Landsat-7 Multispectral
Bands 2,3 and 4*



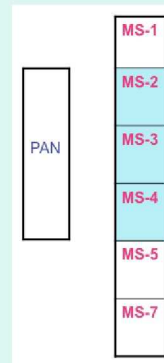
Landsat-7 Panchromatic Band 8



FUSION

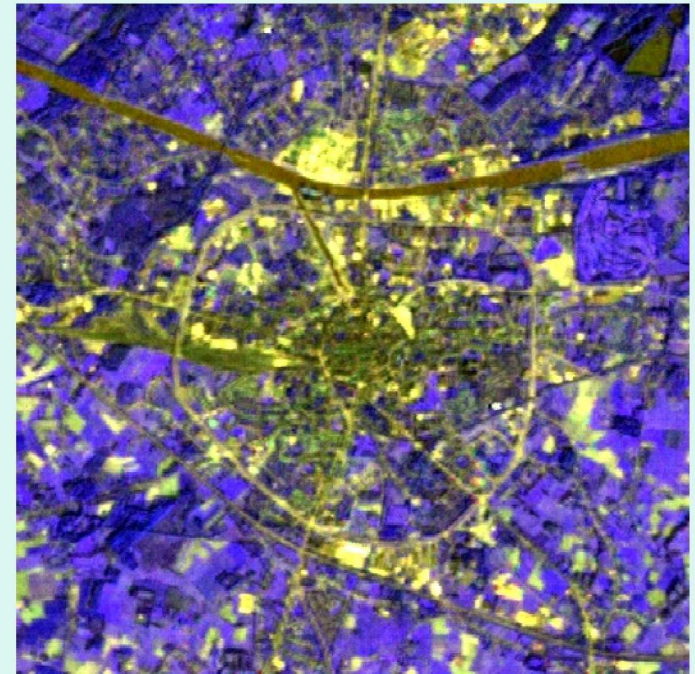
Pan + MS-2	⇒	fused_b2
Pan + MS-3	⇒	fused_b3
Pan + MS-4	⇒	fused_b4

**Spectral Resolution
1 pixel of an MS band**



x1	y1	p1	?
x2	y2	p2	?
x3	y3	p3	ms1
x4	y4	p4	?

*Landsat-7 Pan-Sharpened MS Bands 2,3 and 4
Through Cokriging with Pan Band 8*



Results:

- Correlation: Wavelet: 0.86; PCA: 0.91; Cokriging: 0.92
- Entropy: Wavelet: 3.44; PCA: 3.87; Cokriging: 3.92

References

Jacqueline Le Moigne

- A. Cole-Rhodes, K. Johnson, J. Le Moigne, and I. Zavorin, 2003, "Multiresolution Registration of Remote Sensing Imagery by Optimization of Mutual Information Using a Stochastic Gradient," *IEEE Transactions on Image Processing*, Vol. 12, No. 12, pp. 1495-1511, December 2003.
- J. Le Moigne, N. Netanyahu, and N. Laporte, 2001, "Enhancement of Tropical Land Cover Mapping with Wavelet-Based Fusion and Unsupervised Clustering of SAR and Landsat Image Data," Proceedings of the 8-th SPIE International Symposium on Remote Sensing, Image and Signal Processing for Remote Sensing VII, Vol. #4541, Toulouse, France, September 17-21, 2001.
- J. Le Moigne, W.J. Campbell, and R.F. Crompt, 2002, "An Automated Parallel Image Registration Technique of Multiple Source Remote Sensing Data," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40, No. 8, pp. 1849-1864, August 2002.
- J. Le Moigne and R. Eastman, 2005, "Multi-Sensor Registration for Earth Remotely Sensed Imagery," *published in* R.S. Blum and Z. Liu (eds), *Multi-Sensor Image Fusion and its Application*, Marcel Dekker & CRC Press Publication.
- J. Le Moigne, N. Netanyahu, and R. Eastman, 2010, *Registration for Earth Remote Sensing*, Cambridge University Press, in press (available November 2010).
- N. Memarsadeghi, J. Le Moigne, D. Mount, and J. Morissette, 2005, "A New Approach to Image Fusion Based on Cokriging," FUSION'2005, 8th International Conference on Information Fusion, Philadelphia, Pennsylvania, July 25-28, 2005, Volume 1, pages 622-629.
- N. Memarsadeghi, J. Le Moigne, and D. Mount, 2006, "Image Fusion Using Cokriging," 2006 IEEE International Geoscience and Remote Sensing Symposium, IGARSS'06, Denver, CO, July 2006.
- N. Netanyahu, J. Le Moigne and J. Masek, 2004, "Geo-Registration of Landsat Data by Robust Matching of Wavelet Features," *IEEE Transactions on Geoscience and Remote Sensing*, Volume 42, No. 7, pp. 1586-1600, July 2004.
- H.S. Stone, J. Le Moigne, and M. McGuire, 1999, "Image Registration Using Wavelet Techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, Vol. 21, No.10, October 1999.
- I. Zavorin, and J. Le Moigne, 2005, "On the Use of Wavelets for Image Registration," *IEEE Transactions on Image Processing*, Vol. 14, No. 6, June 2005.

ACRONYMS

- AVHRR: Advanced Very High Resolution Radiometer
- CP or GCP: Control Point or Ground Control Point
- GSHHS: Global Self-consistent Hierarchical High-resolution Shoreline
- GOES: Geostationary Operational Environmental Satellite
- GSFC: Goddard Space Flight Center
- MISR: Multiangle Imaging SpectroRadiometer
- MODIS: MODerate resolution Imaging Spectrometer
- NDVI: Normalized Difference Vegetation Index
- SeaWiFS: Sea-viewing Wide Field-of-view Sensor
- SPOT: Satellite Pour l'Observation de la Terre
- WSU: Wright State University

Image Fusion by Image Compositing

Arthur Goshtasby

Wright State University

Image Registration and Fusion Systems

Image fusion

Combining relevant information in two or more images of a scene into a single highly informative image.

Fusion methods:

- Low-level (pixel-based)
- Mid-level (region-based)
- High-level (description-based)

Image compositing

Cutting and smoothly stitching together parts of different images of a scene into a single highly informative image.

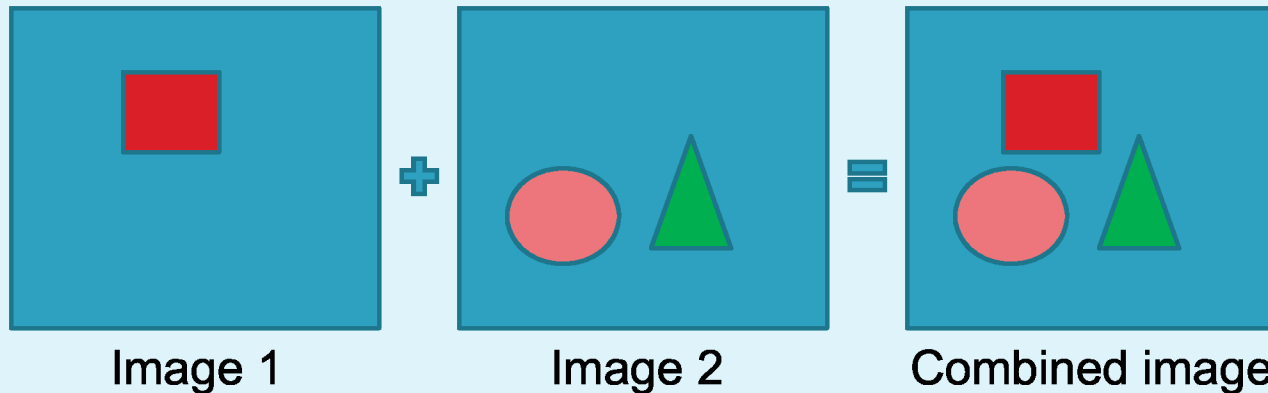
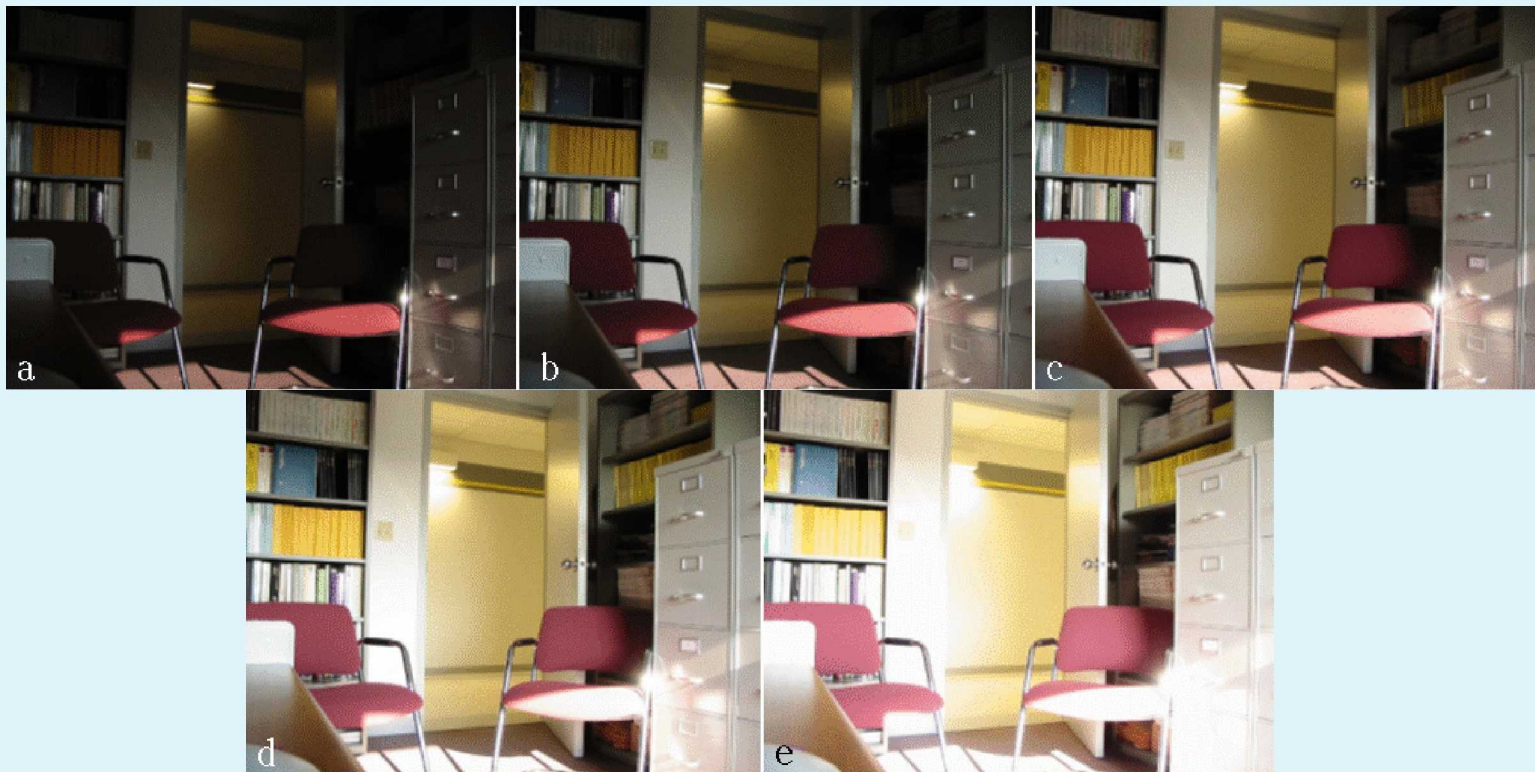


Image fusion as an image blending problem



1. For each local area in the image domain, find the image providing the most information.
2. Cut out the area out of the most informative image.

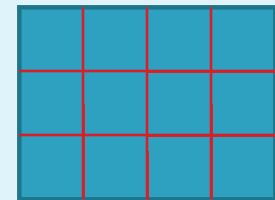
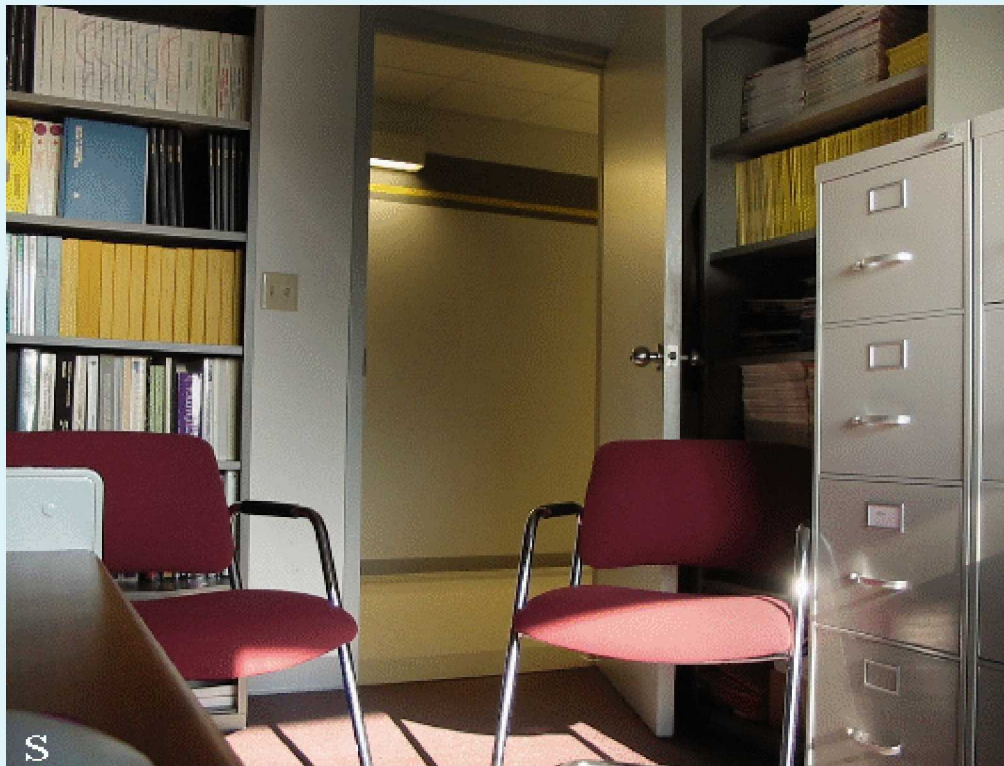


Image domain



Selected local areas in images

3. Blend the local areas to create the fused image.



Selecting the best-block image

- Highest contrast
- Highest entropy

$$E_G = \sum_{i=0}^{255} p_i \log(p_i) \quad \text{Gray-scale}$$

$$E_C = E_r + E_g + E_b \quad \text{Color}$$

The blending function

Assuming (x_i, y_i) represents the coordinates of the center of the i th block, use

$$W_i(x, y) = \frac{G_i(x, y)}{\sum_{j=1}^N G_j(x, y)}$$

as the blending function, where

$$G_i(x, y) = \exp\left\{-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma^2}\right\}.$$

σ is adjusted to the size of the local area.

Image blending algorithm

Given images $\{I_k(x,y): k = 1, \dots, M\}$, and assuming

- 1) the image domain is subdivided into N blocks, and
- 2) for block i , the most informative image is $I_{i.k}(x,y)$,

the fused image is computed from:

$$O(x, y) = \sum_{i=1}^N W_i(x, y) I_{i.k}(x, y)$$

Characteristics of the algorithm

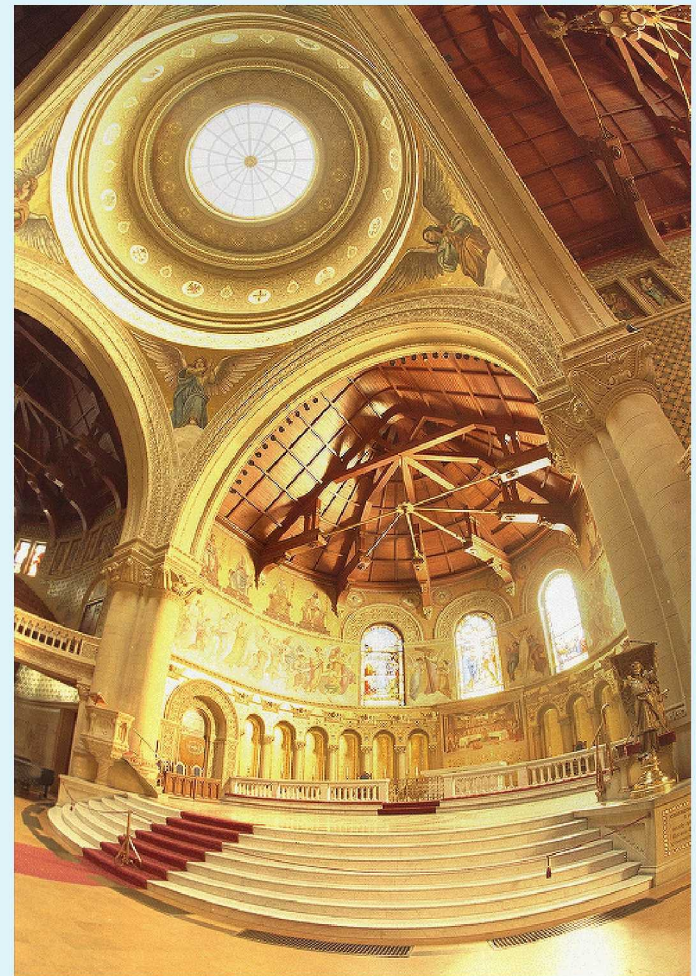
- Parameter σ is adjusted to block size, and block size is chosen based on image size and/or image resolution.
- The blending process does not create information in output that does not exist in input.
- The method is versatile, it can be used to fuse various types of images.

Application 1: Fusion of multi-exposure images

Optimizing criterion: Entropy



Images courtesy of Paul Debevec, USC



Optimizing criterion: Entropy



Original images courtesy of Shree Nayar, Columbia University

Optimizing criterion: Entropy



Original images courtesy of Max Lyons

Application 2:

Fusion of multi-focus images

Original
images



Fused
image

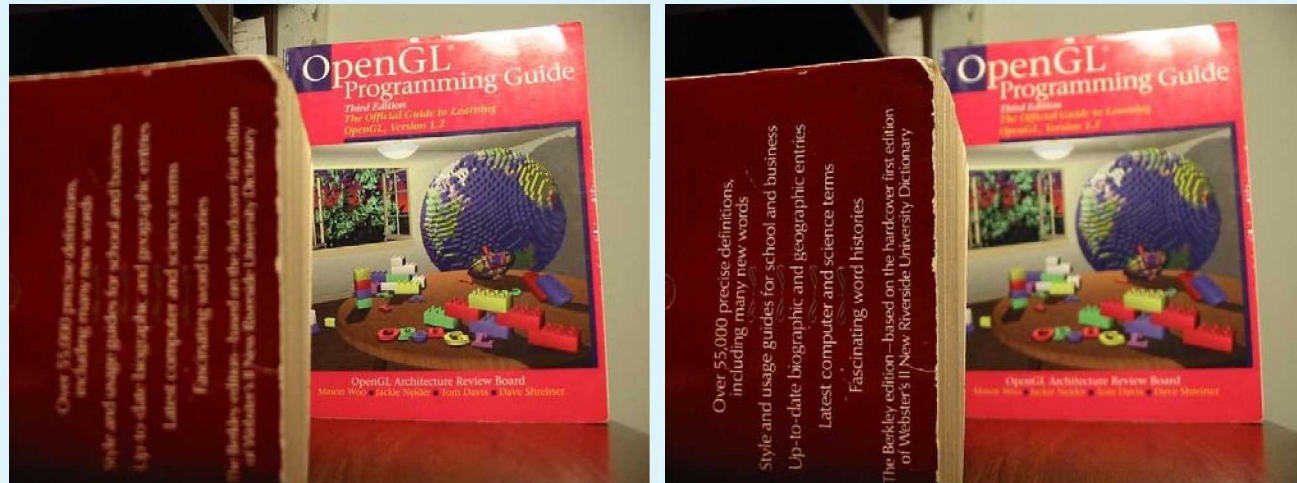


Optimizing criterion:
Contrast

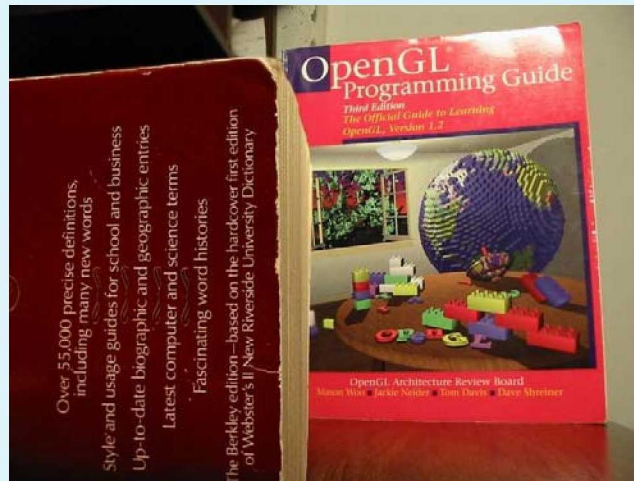
Original images courtesy of Image
Fusion Systems Research

Optimizing criterion: Contrast

Original
images



Fused
image



Concluding remarks

- If the type of information to be maximized is known, this method is guaranteed to produce an image with the highest information content at the desired resolution.
- It can be adapted to image resolution by making block size a function of image resolution.
- By changing the optimization criterion, the method can be used to fuse various types of images.

Image fusion references

1. T. Porter and T. Duff, Compositing digital images, *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, 1984, pp. 253-259.
2. P. Willis, Generalized compositing, *ACM SIGGRAPH*, 2007, pp. 129-135.
3. A. Goshtasby, Fusion of multi-exposure images, *Image and Vision Computing*, vol. 23, 2005, pp. 611-618.
4. A. Goshtasby, Fusion of multifocus images to maximize image information, *Defense and Security Symposium*, 17-21 April 2006, Orlando, Florida, 2006.
5. Special issue of *Information Fusion* on Image Fusion, A. Goshtasby and S. Nikolov (Eds.), vol. 8, 2007.
6. R. S. Blum and Z. Liu, *Multi-sensor Image Fusion and its Applications*, CRC Press, 2005.
7. F. Sroubek and J. Flusser, Registration and fusion of blurred images, *Int'l Conf. Image Analysis and Recognition*, 2004, 124-129.
8. C. Genderen and J. L. van Pohl, Multisensor image fusion in remote sensing: Concepts, methods and applications, *Int'l J. Remote Sensing*, vol. 19, no. 5, 1998, pp. 823-854.

Concluding Remarks

Arthur Ardeshir Goshtasby (WSU)

&

Jacqueline Lemoigne (NASA)

Summary

- Tools and methods for image registration were reviewed.
- Methods for the registration of remotely sensed data at NASA were discussed.
- Image fusion techniques were reviewed.
- Challenges in registration of remotely sensed data were discussed.
- Examples of image registration and image fusion were given.

Conclusions

- A single method cannot register or fuse all image types.
- A custom-made method is require for a particular class of images.
- As the intensity and geometric difference between images increase, the registration process becomes more complex.
- Image fusion necessitates image registration.

Open problems

- An image registration method is needed that can automatically adapt to intensity and geometric differences between images.
- Methods that can verify the correctness of a registration are needed.
- Two main steps in image registration still require further research. They are:
 - Reliable landmark correspondence methods.
 - 2) Image warping models that can adapt to the density and organization of the landmark.

Image registration examples

- Manual registration
- Semi-automatic registration
- Automatic registration
 - Rigid registration
 - Similarity registration
 - Affine registration
 - Projective registration
 - Nonlinear registration

Thank you!